WHEN DOES GEOMETRY ADD TO RECOGNITION? A FUSION BENCHMARK FOR MONOCULAR VOLUME ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Monocular object volume estimation is judged by one error averaged over a dataset, mixing two contributions: (i) the recognised category, which already implies much of an object's volume, and (ii) the geometric pipeline that measures it. A benchmark that cannot separate them cannot say which did the work. We introduce *OneVol*, which factors a monocular estimate into six channels, scores each alone and in combination under rules, combiners, and an oracle on eight datasets, and first asks whether a dataset can support a metric claim. A measurement model gives a bar any estimator must clear, the ground truth's within-class spread, and five validity criteria: five of six volume datasets fail at least one, three on the data alone; and NOCS's 13,371 rows carry the evidence of 46 objects. The same model sizes the next benchmark: about $1/r^2$ targets for a source of within-class correlation r, several hundred to resolve its gain, against the six volume datasets' 9 to 70 targets. The consequence: the median volume of same-class training objects beats the full pipeline in 23 of 24 dataset–backbone cells, and survives a predicted class, unresolved on MADIMA23 and MRGBD. On MetaFood3D, whose classes hold one object each so the prior is only a global median, the best published pipeline beats it. Fused on recognition's terms, the within-class regression keeps the prior and adds only the source's within-class deviation, lowering ECUSTFD's mean log error by 6–11% for every backbone, two of four surviving Holm's correction, and in 25 of 26 cells on the mass datasets, the only ones large enough, confirmed by a pre-registered test on held-out objects and dishes; a second registration, extending the rule to several sources, fails. A weak measurement should be judged by what it adds beyond a strong prior, not by its own accuracy. Code and predictions: `https://github.com/nobody-eh/OneVol-review`.

"All models are wrong, but some are useful."

—George E. P. Box

1 INTRODUCTION

Can a benchmark say what a monocular volume estimate owes to geometry and what it owes to recognising the object? Monocular food volume estimation, which serves dietary assessment and robotic handling, is where it bites: what its reported errors measure depends on the answer. Estimating an object's metric volume from a single image follows a geometric chain: (i) segment the object, (ii) predict depth, (iii) back-project, (iv) integrate. Progress is reported as the mean absolute relative error (ARE) over several datasets, a single number that doesn't say which channel earned it. A second source of information runs through it: the object's category, which the pipeline recovers anyway, since it is also the segmenter's text prompt, and which in most datasets constrains the volume by itself. A reported error therefore conflates two channels that nobody separates, and an improvement cannot be traced to either.

We therefore ask what each dataset can support before asking which method wins, since that depends only on its ground truth. A measurement model, written in log space because volume spans two orders of magnitude and the failure modes are multiplicative (Sec. 3.2), gives a bar any estimator must clear: the within-class spread of the ground truth, computable from the annotations before any method exists. Five validity criteria (Sec. 3.3, swept in App. C.7) then decide whether a dataset's errors may be read as measurements of volume rather than as evidence about recognition. What remains is a fusion problem: the recognition channel, the object's category, offers a strong, cheap,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

nearly unbiased prior; the measurement channel, the geometric chain, a weak, expensive, often badly calibrated estimate of the same quantity; and conventional rules, which average sources or weight them by dispersion, assume similar strength. OneVol collects no new images: it is a protocol, two repaired splits, predictions and a scorer over eight public datasets, six of volume and two of mass (Table 1). Its task is one positive scalar per item, the volume of a food or household object in cm^3 (mass in grams on two datasets), from an image and its class label, which the benchmark supplies so recognition is held fixed rather than evaluated; App. F.4 repeats the analysis with predicted labels. Five findings follow, and App. A records the evidence level behind each.

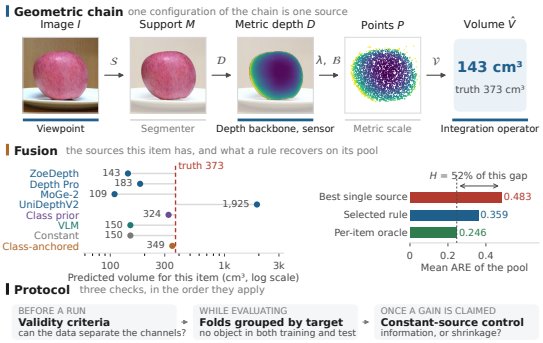

Figure 1: **OneVol as a fusion problem**, for an ECUSTFD apple of 373 cm^3. *Geometric chain*: the stages underlined in blue are the fusion axes drawn here. *Fusion*: this item's sources, and for its pool the best single source, the selected rule and the per-item oracle, with the headroom H. *Protocol*: the three checks, in the order they apply (Secs. 3.3 and 5.4).

1. The benchmarks hold far fewer independent objects than their row counts suggest. On NOCS, 13,371 rows are repeated views of 15 physical objects and carry as much evidence as about 46 independent ones; three of six volume datasets also fail a criterion of the data itself, holding too few objects or classes that already fix the volume (Sec. 5.1). **2. A class-based estimate outperforms the geometric pipeline.** The median volume of same-class training objects outperforms the monocular chain in 23 of 24 dataset–backbone cells, all 24 under symmetric log error, and 64 of ECUSTFD's 70 objects. Calibrated or repaired, it still trails, and so does it against a prior built on predicted rather than looked-up labels (Sec. 5.2). **3. Geometry improves volume estimates when combined with the class-based estimate.** Retaining the class prior and adding only the chain's within-class deviation reduces ECUSTFD's mean log error by 6–11% for all four depth backbones, where it also beats every trained combiner that nests it, and resolves in 25 of the 26 mass-dataset cells large enough to show a gain (Sec. 5.5); a pre-registered test on held-out objects and dishes validates this gain (Sec. 5.6). **4. Simple fixed rules reproduce most of the apparent fusion gain.** Pairing the chain with a VLM mainly corrects gross overestimates toward a plausible scale, a gain a fixed constant reproduces on ECUSTFD (Sec. 5.4). **5. VLMs outperform class-based estimates only when objects in one class differ substantially in size.** Among 66 model–dataset cases with multiple targets per class, a member of a fourteen-model panel beats the class prior's log error in 8, all on the two mass datasets, where within-class variation is greatest (Sec. 5.3).

To study these findings systematically, we make six contributions:

1. **A channel-factorised benchmark** separating recognition from the geometric stages of monocular volume estimation, scored alone and in combination over 37 configurations and a fourteen-member panel (Secs. 3 and 4);
2. **A measurement model and the class-anchored rule** it implies, with a fusion condition that says when a weak source adds information (Sec. 3.2);
3. **An evaluation protocol for repeated observations**: validity criteria, folds and confidence intervals grouped by object, and a constant-source control to distinguish a genuine information gain from shrinkage (Secs. 3.3 and 5.4);
4. **A sample-size rule for the next benchmark**: the targets needed to estimate a source's gain are about one over the square of its within-class correlation, and several hundred are needed to detect that gain reliably, against the 9 to 70 targets the volume benchmarks here hold (simulation in App. F.11);
5. **Two repaired evaluation splits**: NOCS without its laptop annotations, which record a rigid object at 0.6–5% of its own convex hull, and YCB rebuilt from registered infrared depth, 800 items over all 40 objects where one object was usable before (audit and script in App. D.5);
6. **A unified fusion evaluation** of fixed rules, learned combiners, selection, and a per-item oracle over 46 pools using consistent folds (Sec. 5.4).

2 Related Work

The goal of this paper, attributing a volume error to recognition or to measurement, sits between two literatures: benchmarks that report one error per method, and decision-level fusion, which scores a combination against its best member rather than against what recognition supplies for free. **Benchmarks that do not separate the channels.** The closest precedent (full review in App. B) is in single-view 3D reconstruction, where networks mostly act as recognisers: a retrieval baseline matched them (Tatarchenko et al., 2019), exemplifying shortcut learning (Geirhos et al., 2020). Food-volume studies (Dehais et al., 2017; Ando et al., 2019; AlMughrabi et al., 2025; Haroon et al., 2025; Gu et al., 2026) compare pipelines to each other or to a dataset mean, not to what object labels imply, and treat repeated views of one object as independent, so errors are attributable to neither channel. Datasets are inherited rather than audited, and the components they rest on are benchmarked on their own outputs rather than the volume they feed (Ranftl et al., 2022; Alahmari et al., 2024). Benchmark-validity work asks whether a dataset supports a ranking (Torralba & Efros, 2011), not whether it supports a measurement in physical units (Table 4 contrasts these literatures with the needs of channel attribution).

Decision-level fusion, and its unresolved challenges. Each source emits a scalar in the same unit, so classical frameworks apply: (i) untrained rules, which combine estimates without fitting weights (Khaleghi et al., 2013; Meng et al., 2020), (ii) conflict handling, for sources that disagree (Lefevre et al., 2002; Xiao, 2019), (iii) dynamic selection, one source per item (Woods et al., 1997; Ko et al., 2008; Cruz et al., 2018), and (iv) the per-sample oracle, which bounds what selection can reach (Ruta & Gabrys, 2005; Shipp & Kuncheva, 2002), with trained stacks (Granger & Ramanathan, 1984; Breiman, 1996). These approaches seek the most accurate combination of the available estimates and score it against the best of them, not against the estimate recognition supplies for free. Vision–language models are likewise used as sources for their generality (Han et al., 2026), never measured against the baseline of naming the object; a nutrition-estimation survey names portion ambiguity and within-class variability as open problems (Teng et al., 2027). OneVol evaluates the same combinations of volume estimates and fusion rules, but treats the estimates differently: the class-based estimate is a strong baseline available before any geometric processing. We therefore measure each rule's gain against that baseline and against a constant-volume control, group evaluation folds by physical object, and report the best result achievable by selecting a source per item as an upper bound.

3 OneVol as a Fusion Problem

The benchmark rests on the channels an estimate factors into, with a semantic prior and the fusion rules (Sec. 3.1), a measurement model that says what a combination can gain (Sec. 3.2), and the oracle and validity criteria that bound and license every reading (Sec. 3.3).

3.1 Channels, the semantic prior and the fusion rules

A monocular volume estimate is produced from an image I by the geometric chain:

$$\hat{V} \;=\; \mathcal{V}\big(\mathcal{B}\big(\lambda \cdot \mathcal{D}(I, M)\big)\big), \qquad M = \mathcal{S}(I), \tag{1}$$

where $\mathcal{S}$ segments the object into a mask M, $\mathcal{D}$ predicts a depth field on it, the metric scale λ converts that field to metres, $\mathcal{B}$ back-projects the scaled depth to points, and the integration operator $\mathcal{V}$ turns the points into a volume. A segmenter, a depth backbone, a scale cue and an integration operator specify Eq. (1) completely; we call such a pipeline a *source*, since the benchmark sees only the number it returns. Three sources use no geometry: the class prior, the median volume of same-class training objects; a VLM run, the model's own answer; and a constant. OneVol adds a channel the chain does not name: the object category k the recognition stage recovers, which in practice is also the text prompt the segmenter receives. A *channel* is a stage that can be varied on its own, and a *pool* is a set of sources differing along one channel (viewpoint, depth sensor, backbone, operator or support mask), or, in a heterogeneous pool, the chain beside the semantic sources. Fusion maps a pool to one estimate (Fig. 1; vocabulary in App. C.1).

The semantic channel. The class prior estimates the volume of item i as $\hat{V}_i^{\mathrm{sem}} = \mathrm{median}\{V_j : k_j = k_i,\ j \in \mathrm{train}\}$, over the training items j whose class label k_j is that of item i; the median rather than the mean because volumes within a class are right-skewed. It uses recognition alone,

so its error is the best recognition alone can achieve, and a pipeline that does not beat it has not shown its geometry contributes anything. Where a class holds one object, its median is that object's volume, so we fall back to the median over all training items: the global median. Priors are always fitted on folds grouped by physical target (a *target* is a physical object; an *item* is an evaluated row), so views of one object never fall on both sides of a split. Grouping by row instead inflates the prior's apparent accuracy by up to $4.4\times$ where a class prior is learnable (App. E.6).

Fusion rules. We evaluate six rules that fit no parameters: (i) mean, (ii) median, (iii) trimmed mean, (iv) geometric mean, (v) dispersion-weighted mean and (vi) conflict-aware rule, and a learned *selector*, a random forest that predicts which source to trust for an item from cues computed before integration, such as mask area and depth spread. Each pool's rule is chosen by nested selection, so no rule is chosen on its own scored rows. On the same folds we assess trained combiners: a precision-weighted mean, least-squares stacks in cubic centimetres and in log space (Granger & Ramanathan, 1984; Breiman, 1996), a selector reading the source outputs, and three dynamic-selection methods from the literature (Woods et al., 1997; Ko et al., 2008). All are fitted and scored out of fold (App. C.2).

3.2 A MEASUREMENT MODEL, AND THE FUSION IT IMPLIES

Volume is positive and spans two orders of magnitude, and every failure mode we audit is multiplicative: a scale cue wrong by a factor, a depth wrong by a ratio. We therefore model the logarithm of the volume, in which such errors are additive. For item i of class $c(i)$, we write $\log V_i = \mu + \alpha_{c(i)} + \delta_i$ with $\mathrm{Var}(\delta \mid c) = \sigma_w^2$, where μ is the mean log volume over all items and σ_w is the within-class standard deviation that the accuracy bar below is stated in. A source s reports

$$\log \hat{V}_i^{(s)} = a_s + \beta_s \left(\mu + \alpha_{c(i)}\right) + \gamma_s \, \delta_i + \varepsilon_{si}. \tag{2}$$

Here α_c is the class effect, how far objects of class c depart from the overall mean; δ_i the deviation of object i from its own class, and ε_{si} the noise of source s on item i, what remains once the class and the object are accounted for; β_s is how much of the class effect a source reproduces and γ_s how much of the object's own deviation it sees. Two consequences follow, both standard for this model and derived in App. C.3.

The accuracy bar. The class prior predicts $\mu + \alpha_c$, so its log RMSE cannot fall below σ_w: the within-class spread is a floor on what any class-level estimate can reach. Computing it needs neither a model nor a method, so a dataset's author can state in advance the accuracy a volume estimator must surpass to *replace* the prior. A source combined with the prior, rather than replacing it, has a lower threshold to clear.

The class-anchored combination. The model says how to combine the channels: keep the class centre the prior supplies, and add only what the source says about this object relative to others of its class, which is $d_i = \log \hat{V}_i^{(s)} - \overline{\log \hat{V}^{(s)}}_{c(i)}$, its log estimate minus its own class average, which cancels the class terms of Eq. (2) and leaves $\gamma_s \delta_i + \varepsilon_{si}$, the object's deviation as that source sees it. Among the estimates linear in d that keep the prior's class centre $\hat{m}_c$, the one of smallest variance is

$$\log \hat{V}_i^{\mathrm{fus}} = \hat{m}_{c(i)} + b \, d_i, \qquad b = \frac{\mathrm{Cov}(d, \delta)}{\mathrm{Var}(d)}, \tag{3}$$

with population log RMSE $\sigma_w \sqrt{1 - r_{\mathrm{wc}}^2}$, where r_{wc} is the within-class correlation of d and δ. With b known, a source improves on the prior *in combination* whenever $r_{\mathrm{wc}} \neq 0$, regardless of its own error, and the size of the improvement is predicted by one statistic computed on the training folds, the within-class correlation r_{wc}. A constant source has $d = 0$ and returns the prior, so this rule passes the constant-source control of Sec. 5.4 by construction. We estimate $\hat{m}_c$ as the class median and b by least squares within class on the training folds, limiting d to three robust standard deviations so that one degenerate output cannot move the slope; every result is for this fitted, robust variant of Eq. (3).

As an estimator, Eq. (3) is not new: it is within-class (fixed-effects) regression (Mundlak, 1978), a special case of shrinkage towards a group mean (Efron & Morris, 1975). The model's contribution is its interpretation as a fusion rule: clarifying which part of a source to fuse (within-class deviation, not overall level), ensuring robustness against a constant source, and predicting the gain. Estimating

b from n_{eff}, the number of independent targets a dataset effectively provides (Sec. 5.1), in classes with at least two members, costs about $\sigma_w^2/n_{\text{eff}}$, so the rule should help only where

$$r_{\text{wc}}^2 \;>\; 1/n_{\text{eff}}. \tag{4}$$

We call Eq. (4) the *fusion condition*. It requires only a source's within-class correlation on training folds and the number of independent targets, allowing practitioners to assess fusion potential before building a combiner.

3.3 Ceilings, the headroom metric and validity criteria

We score two errors: mean ARE, $A = \frac{1}{n}\sum_i |\hat{V}_i - V_i|/V_i$, and the mean absolute log error, $L = \frac{1}{n}\sum_i |\log \hat{V}_i - \log V_i|$, the *symmetric log error*, with non-positive outputs floored at $1\,\text{cm}^3$. Each pool is bounded by a per-item oracle that selects the source with the smallest relative error. This bounds selection, not combination, so a blend can exceed it. The headroom H of Fig. 1 is the share of the oracle's available gain that a rule captures (App. C.4).

Validity criteria. Each dataset is tested against five criteria before any error is reported. (i) VARIATION: the ground-truth volume varies across items. (ii) TARGETS: at least 30 distinct physical targets. (iii) CLASSES: more than one class. (iv) SPREAD: the class does not determine the volume, screened by the within-class coefficient of variation over distinct targets at a nominal 0.10, applicable only where at least half the classes hold two distinct targets (App. C.7 sweeps both thresholds). (v) SKILL: the chain beats the best prior that can be learned from the labels, the cross-validated class or global median; it describes the pair of benchmark and pipeline rather than the data, so a better chain flips it. Comparing two sources on the same items needs VARIATION and TARGETS; reading an error as metric accuracy requires all five. Where TARGETS fails, the intervals resample too few objects for a significance claim, so we mark them § and report only the sign, a *direction*. The same measurement model also sizes a dataset, and the two thresholds differ: a benchmark can be large enough for the class-anchored rule to help and far too small to prove that it does (Fig. 2c; App. F.11).

4 Datasets and Protocol

We evaluate on ECUSTFD (Liang & Li, 2017), MADIMA23 (Abdur Rahman et al., 2023), MRGBD (Sari & Gofuku, 2023), MetaFood3D (Chen et al., 2024) in its 34-item challenge release, a YCB split (Calli et al., 2015), and a NOCS split (Wang et al., 2019). For the VLM panel, we add Nutrition5k (Thames et al., 2021) and OmniFood8K (Yu et al., 2026), which use mass in grams as ground truth and whose class is the heaviest ingredient where it holds at least 60% of the dish's mass, computed from the published ingredient masses and so not available to a method at test time; these enter through the semantic channel and are never pooled with the volume datasets (Table 1). A target is identified by its distinct ground-truth value, so objects sharing a volume are grouped (78 objects behind ECUSTFD's 70 targets), which errs towards wider intervals. Rows greatly outnumber targets in four datasets, by 12 views per target on ECUSTFD and 891 on NOCS. Class membership tightly constrains the target on the food benchmarks, but less so on NOCS and the mass datasets.

Sources and scale. Each configuration of Eq. (1) uses published components. The support $\mathcal{S}$ is a GroundingDINO proposal (Liu et al., 2024) prompted with the class label, refined by the Segment Anything Model (Kirillov et al., 2023). The depth $\mathcal{D}$ is from one of four published metric monocular backbones, ZoeDepth (Bhat et al., 2023), Depth Pro (Bochkovskii et al., 2025), UniDepthV2 (Piccinelli et al., 2026) and MoGe-2 (Wang et al., 2025a), or, on MADIMA23's sensor axis, from the dataset's RGB-D capture, and the operator $\mathcal{V}$ is one of four integrations of the back-projected points, the convex hull (*hull*) among them. The *reference chain* for each dataset is a configuration set before the study. One rule sets λ for every configuration, with no constant fitted to the evaluation target: the confidence-weighted median of the cues a capture affords, (i) a reference object such as the ECUSTFD coin, (ii) sensor depth, (iii) the support plane and (iv) a size prior (App. D.2). One command scores a new method from a CSV of per-item estimates, returning the criteria, the paired comparison with the prior and the class-anchored gain (App. D.10). App. D.11 is the benchmark card: (i) motivation, (ii) composition, (iii) known defects, (iv) intended use, (v) licence and (vi) maintenance, each pointing to where the paper establishes it. All eight datasets are public, and none is redistributed here; repeated views of one object are not independent, so every interval is a percentile bootstrap clustered by target (Table 11; App. C.6).

Table 1: **What the eight datasets contain, and what each can support.** *Targets* counts distinct physical objects and *Views* is rows per target; the within-class CV is the median over multi-target classes, with that count in parentheses. MRGBD's 12 objects resample as 9 targets, three of its volumes recurring across class pairs. Ranges and medians in cm^3; † mass in grams. *Fails* lists the validity criteria of Sec. 3.3 a dataset does not meet (–, SPREAD not applicable): ‡, SKILL evaluated with the incumbent VLM (Qwen2.5-VL-7B) as the pipeline; *, SKILL passes on mean ARE by a difference whose interval spans zero and fails on the log error.

Dataset	Domain	Rows	Targets	Views	Cls.	Targets per cls.	Range	Median	Within-cls. CV	Fails	Licensed use
ECUSTFD	single foods	837	70	12	19	3.7	39–373	155	0.098 (18/19)	SPREAD, SKILL	comparisons
MADIMA23	plated meals	82	36	2	34	1.1	37–639	105	0.073 (2/34)	SKILL	comparisons
MRGBD	single foods	191	12	16	5	2.4	70–175	120	0.053 (5/5)	TARGETS, SPREAD, SKILL	directions
MetaFood3D	plated foods	34	34	1	34	1.0	5–828	91	–	none*	comparisons, accuracy
YCB	household objects	800	40	20	40	1.0	8–2,630	190	–	SKILL	comparisons
NOCS-REAL275	household objects	13,371	15	891	5	3.0	147–1,012	301	0.332 (5/5)	TARGETS, SKILL	directions
Nutrition5k†	plated dishes	1,347	1,347	1	49	27.5	5–871	96	0.472 (49/49)	SKILL‡	semantic channel
OmniFood8K†	plated dishes	8,036	8,036	1	267	30.1	11–2,445	178	0.274 (183/267)	SKILL‡	semantic channel

5 RESULTS

The results follow the argument: what the benchmarks can support (Sec. 5.1), what recognition alone is worth, as a lookup and as a VLM panel (Secs. 5.2 and 5.3), what fusion reaches (Sec. 5.4), what anchoring on the prior adds (Sec. 5.5), and whether it replicates (Sec. 5.6).

5.1 WHAT THE BENCHMARKS CAN SUPPORT

We first ask whether each benchmark holds enough independent targets and within-class variation to carry a claim. Five of six volume datasets fail at least one criterion, three of them a criterion of the data itself (Table 1). MRGBD and NOCS fail TARGETS, with 12 and 15 objects behind 191 and 13,371 rows, and ECUSTFD fails SPREAD at 0.098 against the nominal 0.10. Nutrition5k and OmniFood8K pass the four data criteria, with $1{,}347$ and $8{,}036$ targets, in grams rather than cm^3. Clustering the bootstrap on the target rather than the row widens NOCS's interval from $[2.90, 3.70]$ to $[0.87, 7.34]$, an inflation of $8.1\times$ growing with view multiplicity. Kish's design effect (Kish, 1965) prices the rows (Table 12): NOCS's 13,371 carry the evidence of 46 (26–141), and ECUSTFD's twelve views per target are worth about four. Adding NOCS's 15 admissible training objects takes it past TARGETS, and the prior stays ahead ($+0.379$, CI $+0.153$ to $+0.569$; App. E.3).

5.2 THE SEMANTIC CHANNEL AGAINST THE GEOMETRIC CHAIN

We then compare the class prior with the full geometric chain. The reference chain is (i) ZoeDepth with the hull operator on ECUSTFD, MetaFood3D, YCB and NOCS, (ii) Depth Pro on MADIMA23 and (iii) MoGe-2 on MRGBD. It is the weakest backbone on ECUSTFD and NOCS and the strongest on MADIMA23 and MetaFood3D, so nothing rests on it (Table 19). The prior is the more accurate side in 23 of the 24 cells and in all 24 under the symmetric log error. Against the best backbone, the prior's margin is $\times 8.9$ on ECUSTFD, $\times 1.4$ on MRGBD and MADIMA23, $\times 7.3$ on YCB and $\times 3.7$ on NOCS. Nor is the chain a straw man (Apps. F.2 and F.3): (i) given one fitted factor, it still trails the prior in 23 of the 24 cells, (ii) the ECUSTFD chain with its coin defect repaired loses on 65 of 70 targets, and (iii) two trained opponents draw level with the prior on most datasets and lead only on YCB, where no class can be looked up.

On ECUSTFD, the prior's win holds target by target and under log error, but in mean ARE one item dominates. We fixed the primary family, its metric, and its correction before the paired analysis, though we did not pre-register it (App. F.1). ECUSTFD's percentile interval excludes zero ($+6.334$), MADIMA23's does not, and MetaFood3D is unresolved (Table 1). One item, a saturated coin estimate that turned a doughnut into more than a cubic metre, carries 96% of the ARE difference, so the cluster-robust t interval spans zero; the target-level evidence does not, the prior winning 64 of 70 targets ($p = 5 \times 10^{-6}$), $+0.266$ $[+0.178, +0.357]$ without the item. Under the symmetric log error, a declared sensitivity analysis shows that every dataset in that family survives; MetaFood3D is the one reversal, and only on mean ARE (Table 1).

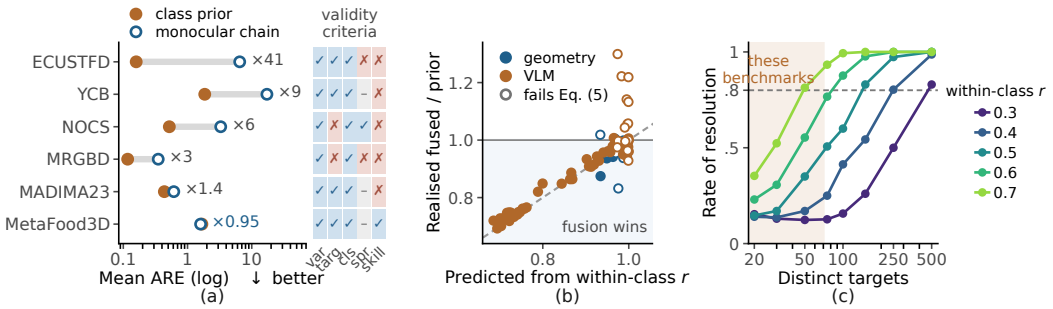

Figure 2: **(a) A prior with no image geometry against the full monocular chain.** Mean ARE per volume dataset, log scale, with the validity criteria of Sec. 3.3 beside each row. The largest margins are on datasets that cannot support a metric claim, and MetaFood3D, the one closest to supporting one, is where the chain leads; ECUSTFD's $\times 41$ is $\times 2.7$ without its one degenerate item. **(b) What the model predicts, and where it is right.** Predicted log-RMSE ratio of fused to prior, $\sqrt{1 - r_{\text{wc}}^2}$, against the realised out-of-fold ratio per cell ($r = 0.87$ over 82 cells; 0.99 mass, 0.45 volume); below 1 fusion beats the prior, open markers fail Eq. (4). **(c) How large a benchmark would have to be.** Simulated rate of resolving the gain, in either direction, against the number of distinct targets, at five per class; the band marks the 9 to 70 these datasets hold (App. F.11).

Three further checks bound the claim. The prior above is a lookup under perfect recognition: queried with the *incumbent* VLM's name, it degrades on ECUSTFD from 0.159 to 0.346 in ARE, still an order of magnitude ahead of the chain, and falls behind it on MADIMA23 and MetaFood3D (App. F.1). A classifier does better: a logistic head on frozen image features, fitted on the same folds, recovers 96, 95 and 81% of the labels on ECUSTFD, MRGBD and NOCS, and the prior built on its predictions keeps the chain behind by 0.252, 0.133 and 0.432 in log error, though on MRGBD and MADIMA23 it no longer resolves (App. F.4). On MetaFood3D and YCB, the prior is the global median, so a label change does nothing; MetaFood3D is also the only volume dataset that passes every applicable criterion, and its geometry wins. The claim does not rest on our chain: against the best published method each dataset has, the prior is ahead on ECUSTFD by 0.277 in mean ARE $[+0.170, +0.401]$, where that method is sixteen times more accurate than our chain, and behind on MetaFood3D by 0.923 (App. F.9).

5.3 VLMs help where targets vary within a class

We next test whether a VLM adds information beyond the class prior, and where. The panel spans seven vision encoders, three language-model families and 2B to 235B parameters on identical crops (Table 40). On ECUSTFD, scored per row, within-class correlation runs from 0.02 to 0.11, and no member beats the class prior alone or gains over it in combination. On plated portions this inverts: all 8 of the 66 cases where a member beats the prior's log error fall there, and on Nutrition5k the correlation is 0.65–0.73 for six of the eleven 16-bit members, the best reaching an ARE of 0.44 against the prior's 0.55. What they lack is calibration, not information: median bias runs from $\times 0.67$ to $\times 1.50$. One number reconciles the two groups, the median within-class spread of the ground truth is $\times 1.08$ on MRGBD, $\times 1.12$ on ECUSTFD, $\times 1.40$ on OmniFood8K and $\times 1.68$ on Nutrition5k. They differ in target, scene and prompt too, so the design bounds the change without isolating it. A probe of the cached answers finds no excess at the true value (App. G.2).

5.4 Fusion ceilings, and information against shrinkage

We next test whether gains from combining sources reflect information or merely shrinkage toward a plausible value. Selection helps, and the selected rules recover little. For each of the 46 pools we report (i) the best single source, (ii) the selected rule, (iii) the trained combiners and (iv) the per-item oracle (Table 49). The oracle can remove 17–63% of the best source's error on the backbone pools and 37–76% across all channels. Pairing the chain with the VLM removes 5–45% of the best member's error on four datasets and makes MetaFood3D and NOCS 24 and 13% worse.

Over the 46 pools, with datasets weighted equally, one combiner is clearly better (Table 50): the least-squares stack fitted in log space, where these errors live, lowering error by 46% (95% interval 28–61%) and improving 34 pools. Its intercept is a fitted constant, and that constant alone gains 24%

Table 2: **Fusing the prior with a source's within-class deviation.** Mean ARE of the prior and of the fused estimate; gain in mean absolute log error (prior minus fused; **bold**: the target-clustered interval excludes zero on a TARGETS-passing dataset); permutation p against the source exchanged between whole targets of one class († survives Holm's correction over the 16 geometry cells); random target partitions won, of 30; log-RMSE ratio predicted / realised. Top: each dataset's reference chain. Bottom: the panel member with the median gain, scored on the items it covers, so its prior differs from the chain's. Full notes in App. I.2. §, fails TARGETS.

		Mean ARE↓					
Dataset	**Source**	**Prior**	**Fused**	**Log-error gain [95% CI]**	**Perm. p**	**Partitions won**	**RMSE ratio pred. / real.**
ECUSTFD	geometry, ZoeDepth	0.159	0.146	**+0.016** [+0.007, +0.026]	0.0024†	30/30	0.94 / 0.94
MADIMA23	geometry, Depth Pro	0.436	0.353	+0.088 [-0.010, +0.183]	0.5263	30/30	0.98 / 0.83
MRGBD§	geometry, MoGe-2	0.117	0.105	+0.014 [+0.005, +0.027]	0.091	30/30	0.93 / 0.87
MetaFood3D	geometry, ZoeDepth	1.673	1.682	+0.001 [-0.100, +0.101]	0.1508	21/30	– / 0.97
YCB	geometry, ZoeDepth	1.854	1.804	+0.058 [-0.049, +0.165]	0.0004†	30/30	– / 0.93
NOCS§	geometry, ZoeDepth	0.521	0.499	+0.015 [+0.002, +0.028]	0.099	30/30	0.98 / 0.98
ECUSTFD	Qwen3-VL-32B (NF4)	0.145	0.145	+0.001 [-0.003, +0.005]	0.361	19/30	1.00 / 0.98
NOCS§	Llama-3.2-90B-V	0.502	0.486	+0.006 [-0.021, +0.035]	0.129	14/30	0.94 / 0.96
Nutrition5k	Qwen2.5-VL-7B	0.545	0.388	**+0.110** [+0.093, +0.127]	≤ 0.001	30/30	0.76 / 0.76
OmniFood8K	Llama-3.2-90B-V	0.273	0.212	**+0.032** [+0.028, +0.036]	≤ 0.001	30/30	0.75 / 0.76

(-6 to 49%), so the best source is often worse than a constant; against the constant, the stack still lowers error by 28% (13–43%) on 40 of the 46 pools, so most of the gain comes from the sources. Fitted in cubic centimetres instead, the same stack only ties the best source (-10%).

Control sources. We pair the chain with a constant equal to the VLM's modal answer (Table 51). Where the VLM helps, the constant pairing's error is 0.78–0.91 of the VLM pairing's, and on ECUSTFD, where the constant stays within a post hoc 10% margin, it is the better partner under the log error (one-sided $p = 0.0003$, surviving Holm's correction). Pairing with the prior beats the constant by 0.148 on ARE and 0.156 on log error, both intervals excluding zero: a constant reproduces the VLM's contribution, not the prior's. Fusion gains evidence only if they beat a constant-source control.

5.5 FUSING GEOMETRY ON RECOGNITION'S TERMS

We next test whether geometry, weak as a standalone estimator, still adds within-class information beyond the prior. No linear combination of the geometric sources *replaces* the prior on ECUSTFD or NOCS: over three nested families of bias-corrected combinations fitted in sample, none reaches it, even with free weights and an intercept (Table 61).

Geometry does improve on the prior *in combination*, under the class-anchored rule of Eq. (3) (Table 2). When a benchmark is large enough to settle the question, it does: 25 of the 26 cells meeting the sample size of App. F.12, all on the mass datasets, show a gain. On Nutrition5k, all fourteen panel members beat the prior against their within-class shuffles; 13 had intervals excluding zero, and the median member lowered mean ARE from 0.545 to 0.388 (exploratory; unchanged when the dish class is predicted from the image, App. F.8). The volume datasets point the same way but hold too few targets to settle it. On ECUSTFD, keeping the prior and adding the chain's within-class deviation lowers the mean absolute log error by 0.016 (95% CI 0.007–0.026) and wins all 30-fold partitions. Every backbone beats its *within-class shuffle*, its own values exchanged between targets of one class ($p = 0.002$–0.011), but only two of them, the reference chain and MoGe-2, survive Holm's correction over the 16 geometry cells passing TARGETS ($p = 0.024$ and 0.020; Table 63). The gain is small because ECUSTFD's classes barely vary: the model predicts a log-RMSE ratio of 0.94 for the reference chain, and 0.94 is realised out of fold. On MADIMA23, the gain is larger ($+0.088$) but indistinguishable from its within-class shuffle ($p = 0.53$).

What helps is the constraint: with the same features and folds, four alternatives do worse on ECUSTFD, for every backbone (Table 65): (i) a log-space stack of prior and chain; (ii) the Mundlak regression, which nests both that stack and the rule (Mundlak, 1978); (iii) a random-intercept model with empirical-Bayes class effects; and (iv) gradient boosting. The stack is worse by 0.014–0.022 and the Mundlak regression by 0.017–0.029 in every interval, excluding zero; the other two lose 0.04–0.07. The chain is poorly calibrated on ECUSTFD but ranks objects within a class: the rule

takes calibration from the prior and ranking from the chain, while a combiner free to re-weight the prior pays in variance.

Fig. 2b sets the predicted log-RMSE ratio against the realised one over the 82 (dataset, source) cells whose classes hold at least two targets. They correlate at $r = 0.87$, but the mass datasets carry it, $r = 0.99$ over their 28 cells against 0.45 over the volume datasets' 54. The formula cannot predict harm, since $\sqrt{1 - r_{\mathrm{wc}}^2} \leq 1$, yet the rule loses to the prior in 14 of those 54; the fusion condition flags 13 of the 14, and of the 17 volume cells passing it, one does harm. Swept over 18,638 refits on the mass datasets, the rule harms in 47% where the condition fails and 5.6% where it passes (Table 66). App. I.9 gives the variance counterpart; the slope transfers between datasets and the gain widens as the class coarsens (App. F.10).

5.6 A PRE-REGISTERED CONFIRMATION, AND A REGISTRATION THAT FAILED

Because the class-anchored results above are exploratory, we chose the rule and its tuning on the data they are scored on; we test whether the gain replicates under a registered plan. We registered the hypotheses, samples, tests and criteria as a plan frozen in a public, server-timestamped release, not at a third-party registry, before running the chain on two untouched samples (App. I.10): (i) the 81 objects of ECUSTFD's `trainval` split, disjoint from the test split above, 604 side views over 74 targets, and (ii) the 1,741 Nutrition5k dishes outside the paper's 1,347, of which 1,738 have images.

Both primary hypotheses are confirmed (Table 62): on the held-out objects the chain's within-class deviation lowers the prior's mean log error by 0.034 (95% CI 0.005–0.076; $p = 0.016$), and three of four backbones survive Holm's correction; on the held-out dishes every source does, at the floor of 100,000 draws, gaining about 0.009–0.010. That interval contains the exploratory estimate of 0.016. Against the combiners that nest it, the result splits by sample size: on 74 ECUSTFD targets the rule beats the log stack and the Mundlak regression; on 1,738 dishes it only ties the latter.

A second registration failed: extended to one slope per source on the top views of the held-out objects, the rule does not beat its single-source form (-0.004, CI -0.016 to $+0.005$, $p = 0.64$), so the paper's claim rests on fusing one source's deviation (App. I.11).

6 DISCUSSION AND LIMITATIONS

Apparent gains can come from calibration or shrinkage, and information is recovered most reliably by anchoring on the strong prior. Three lessons follow: (i) fit a combiner where its errors live, (ii) anchor on a strong prior rather than pool with it, and (iii) read every gain against a constant. They should hold wherever a strong prior sits beside a weak measurement, and do on a problem with no image or volume (Apps. F.5 and J.2).

Limitations. Only MetaFood3D passes every applicable criterion, on 34 items, so the strongest statements about volume concern evaluation. The confirmed gain is small, 13% and 3% on the held-out objects and dishes, about 2 cm^3 on a median 155 cm^3 object (App. F.6): attribution, not a deployable improvement. The four backbones share a segmenter, scale rule and operator, so their agreement is one test, not four. Simulation puts every volume benchmark here below the size needed to resolve such a gain at 80% power (Apps. J.4 and F.11).

7 CONCLUSION

We introduced a benchmark separating the semantic prior from geometric measurement. Recognition explains much of what is attributed to geometry, so these benchmarks fail discriminant validity (Jacobs & Wallach, 2021), with their rows overstating the evidence by up to two orders of magnitude. Anchored on the prior, geometry still adds within-class information, confirmed by a pre-registered test: a weak measurement should be judged by what it adds beyond a strong prior, and a benchmark by whether it is large enough to tell. What follows for practice is concrete: a dataset can be sized before it is collected, several hundred targets at five or more per class against the 9 to 70 here; a method can be scored against the prior and a constant in one command, which reports the criteria, the gain and the targets needed to resolve it; and a gain a benchmark cannot resolve should be reported as a direction, not as a number.

AI USE STATEMENT

In this work, we used a generative AI tool, Claude (Anthropic), for the specific tasks listed here: implementing methods, designing the robustness experiments named below, and drafting the reading of their results. It helped write the analysis scripts that compute the reported statistics and generate the tables and figures. At the authors' request it also proposed, implemented and ran the robustness analyses added during internal review, namely the comparison with a calibrated and repaired chain and the two trained opponents (App. F.2 and F.3), the sweep of the fusion condition (App. I.8), the memorisation probe (App. G.2) the extension of NOCS with its training split (App. E.3) and the prior rebuilt on predicted labels (App. F.4), each with its plan written down before the first run, and it drafted the reading of their results. We did not use generative AI tools to generate synthetic data, develop the measurement model, formulate mathematical claims, provide proof ingredients or write proofs, or propose the paper's registered hypotheses; translation assistance, dataset cleaning and reformatting, and qualitative data analysis do not apply to this work. Additionally, we used the same tool to edit software code, draft and edit text for readability and length, suggest the paper's structure, conduct mock reviews of drafts, and write the plotting code for Fig. 1 and Fig. 2b. A generative model produced no image: every figure is a plot that scripts draw from the released data, and the photographs are frames from the published dataset. The vision–language models of Sec. 5.3 are objects of study, not research tools: their outputs are the data the benchmark scores. We have reviewed all AI-assisted work: the authors reviewed and edited the AI-assisted text and analyses, and every reported number regenerates from the released per-item predictions with the released scripts. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The human reference in App. F.17 comprises seventeen participants, an online audience drawn from two universities, recruited by an open invitation, who estimated the volume of food in published photographs through an anonymous online form (Google Forms). Participation was voluntary and uncompensated. Before starting, each participant was told the exercise was part of a research study on food-volume estimation, that their anonymous answers would be used in a research publication, and that they gave consent through the form. It collected only numeric volume estimates, with no demographic information or identifiers, and involved no intervention, deception, or risk beyond everyday life. Because the form was anonymous, we cannot determine whether any participant is an author or a member of the authors' groups, and we did not record how many people the invitation reached. We did not record participants' sex or gender, so we do not analyse the human reference by sex or gender; we note this as a limit on how far the crowd's accuracy generalises. The released file contains only a sequential participant number and the estimates; we strip the form's response timestamps, which could serve as weak quasi-identifiers, before release. The images shown are published frames from MetaFood3D (Chen et al., 2024) and depict food only.

REPRODUCIBILITY STATEMENT

The OneVol code, validity criteria, fusion-operator library and the scripts that generate every table and figure are available in the anonymised supplementary material, as a snapshot without history. The registered plan, the held-out sample lists and the frozen analysis code are the release `prereg-v2` of the same repository, published before the held-out pipeline ran; the plan for the multi-source registration of Sec. 5.6, the one that did not replicate, is the release `prereg-v3`, published before the runs it governs. The registration is a server-timestamped release of the authors' own repository, made on 15 and 16 September 2026, not a third-party registry. Because those releases cannot be opened without identifying the authors, the supplementary material holds each plan exactly as frozen, anonymised, with the SHA-256 of every file as committed, so that the public releases can be checked against them afterwards; a first freeze, `prereg-v1`, named an external registry that was never used, and differs from `prereg-v2` only in the tag name and that mechanism. The per-item predictions, scale reports, analysis tables and logs behind every reported number are attached to the supplementary snapshot and to a tagged release of the same repository, whose manifest lists every file with its checksum and maps each table and figure to its source. The supplementary snapshot includes the code, the per-item predictions for every source, and the analysis tables computed from them. From a clean copy, one command scores a prediction file, the analysis scripts recompute their tables from the per-item predictions, and `fusion.tables` formats every

table of the paper from those results (App. D.10). Our code, predictions and analyses are licensed under CC BY 4.0. The release holds per-item predictions, scale reports, analysis tables and logs; the datasets' images and meshes and the evaluated models' weights are not redistributed, and the ground-truth values and item identifiers that the per-item files repeat remain under their publishers' terms (Table 11; App. D.11). The eight datasets are available from their publishers and are cited in the text. The reference implementation is described in App. D.4, the scale protocol in App. D.6, the prompts in App. G.4, the metrics and intervals in App. C.6, the registered plan and its deviations in App. I.10, and the compute in App. D.12. Because we release per-item predictions, every table and figure here regenerates on a CPU in about an hour.

REFERENCES

Lubnaa Abdur Rahman, Ioannis Papathanail, Lorenzo Brigato, and Stavroula Mougiakakou. A comparative analysis of sensor-, geometry-, and neural-based methods for food volume estimation. In *Proc. 8th Int. Workshop on Multimedia Assisted Dietary Management (MADiMa), ACM Multimedia*, pp. 21–29, 2023. doi: 10.1145/3607828.3617794.

Saeed S. Alahmari, Michael Gardner, and Tawfiq Salem. Segment anything in food images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR) Workshops*, pp. 3715–3720, 2024.

Ahmad AlMughrabi, Umair Haroon, Ricardo Marques, and Petia Radeva. VolTex: Food volume estimation using text-guided segmentation and neural surface reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR) Workshops*, pp. 450–457, 2025.

Ahmad AlMughrabi, Guillermo Rivo, Carlos Jiménez-Farfán, Umair Haroon, Farid Al-Areqi, Hyunjun Jung, Benjamin Busam, Ricardo Marques, and Petia Radeva. Benchseg: A large-scale dataset and benchmark for multi-view food video segmentation. *arXiv preprint arXiv:2601.07581*, 2026.

Yoshikazu Ando, Takumi Ege, Jaehyeong Cho, and Keiji Yanai. DepthCalorieCam: A mobile application for volume-based food calorie estimation using depth cameras. In *Proc. 5th Int. Workshop on Multimedia Assisted Dietary Management (MADiMa), ACM Multimedia*, pp. 76–81, 2019.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. doi: 10.48550/arXiv.2502.13923.

J. M. Bates and C. W. J. Granger. The combination of forecasts. *Operational Research Quarterly*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth Pro: Sharp monocular metric depth in less than a second. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025. arXiv:2410.02073.

Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996. doi: 10.1007/BF00117832.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015.

Yuhao Chen, Jiangpeng He, Gautham Vinod, Siddeshwar Raghavan, Chris Czarnecki, Jinge Ma, Talha Ibn Mahmud, Bruce Coburn, Dayou Mao, Saeejith Nair, Pengcheng Xi, Alexander Wong, Edward Delp, and Fengqing Zhu. MetaFood3D: 3D food dataset with nutrition values. *arXiv preprint arXiv:2409.01966*, 2024.

Gerda Claeskens, Jan R. Magnus, Andrey L. Vasnev, and Wendun Wang. The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, 32(3):754–762, 2016. doi: 10.1016/j.ijforecast.2015.12.005.

Rafael M. O. Cruz, Robert Sabourin, and George D. C. Cavalcanti. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018. doi: 10.1016/j.inffus.2017.09.010.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pp. 303–312, 1996.

Joris Dehais, Marios Anthimopoulos, Sergey Shevchik, and Stavroula Mougiakakou. Two-view 3d reconstruction for food volume estimation. *IEEE Transactions on Multimedia*, 19(5):1090–1099, 2017.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104. IEEE, 2025.

Bradley Efron and Carl Morris. Data analysis using Stein's estimator and its generalizations. *Journal of the American Statistical Association*, 70(350):311–319, 1975. doi: 10.1080/01621459.1975.10479864.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, pp. 2366–2374, 2014.

Mihai Cristian Florea, Anne-Laure Jousselme, Éloi Bossé, and Dominic Grenier. Robust combination rules for evidence theory. *Information Fusion*, 10(2):183–197, 2009. doi: 10.1016/j.inffus.2008.08.007.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *ICML*, pp. 2151–2159, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pp. 3828–3838, 2019.

Kristen B. Gorman, Tony D. Williams, and William R. Fraser. Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus Pygoscelis). *PLoS ONE*, 9(3):e90081, 2014. doi: 10.1371/journal.pone.0090081.

Alexandros Graikos, Vasileios Charisis, Dimitrios Iakovakis, Stelios Hadjidimitriou, and Leontios Hadjileontiadis. Single image-based food volume estimation using monocular depth-prediction networks. In *International Conference on Human-Computer Interaction*, pp. 532–543. Springer, 2020.

Clive W. J. Granger and Ramu Ramanathan. Improved methods of combining forecasts. *Journal of Forecasting*, 3(2):197–204, 1984. doi: 10.1002/for.3980030207.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783.

Songen Gu, Lina Liu, Binjie Liu, Sanyi Zhang, Lei Yang, and Yanwei Fu. MonoBite: Scale-aware 3D reconstruction and volume estimation from monocular multi-food images. In *Proc. Chinese Conf. Pattern Recognition and Computer Vision (PRCV)*, pp. 32–46. Springer, 2026. doi: 10.1007/978-981-95-5737-0_3.

Xiaofeng Han, Shunpeng Chen, Zenghuang Fu, Zhe Feng, Lue Fan, Dong An, Changwei Wang, Li Guo, Weiliang Meng, Xiaopeng Zhang, Rongtao Xu, and Shibiao Xu. Multimodal fusion and vision–language models: A survey for robot vision. *Information Fusion*, 126:103652, 2026. doi: 10.1016/j.inffus.2025.103652.

Umair Haroon, Ahmad AlMughrabi, Ricardo Marques, and Petia Radeva. VolE++: A text-guided point-cloud framework for food 3D reconstruction and volume estimation. In *Proc. 21st Int. Conf. Computer Analysis of Images and Patterns (CAIP)*, pp. 386–397. Springer, 2025. doi: 10.1007/978-3-032-04968-1_33.

David I. Harvey, Stephen J. Leybourne, and Paul Newbold. Tests for forecast encompassing. *Journal of Business & Economic Statistics*, 16(2):254–259, 1998. doi: 10.1080/07350015.1998.10524759.

Allison M. Horst, Alison P. Hill, and Kristen B. Gorman. palmerpenguins: Palmer Archipelago (Antarctica) penguin data, 2020. R package version 0.1.0.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 375–385, 2021.

W. James and Charles Stein. Estimation with quadratic loss. In *Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 361–379. University of California Press, 1961.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *NeurIPS*, pp. 5574–5584, 2017.

Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013. doi: 10.1016/j.inffus.2011.08.001.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 4015–4026, 2023.

Leslie Kish. *Survey Sampling*. John Wiley & Sons, New York, 1965.

Albert H. R. Ko, Robert Sabourin, and Alceu Souza Britto, Jr. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008. doi: 10.1016/j.patcog.2007.10.015.

Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the iBims-1 dataset. *Computer Vision and Image Understanding*, 191:102877, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *ICML*, pp. 5637–5664, 2021.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 7, pp. 231–238, 1995.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, pp. 6402–6413, 2017.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: Insights and future directions. *arXiv preprint arXiv:2408.12637*, 2024. doi: 10.48550/arXiv.2408.12637.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

Eric Lefevre, Olivier Colot, and Patrick Vannoorenberghe. Belief function combination and conflict management. *Information Fusion*, 3(2):149–162, 2002. doi: 10.1016/S1566-2535(02)00053-2.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

Yanchao Liang and Jianhua Li. Computer vision-based food calorie estimation: dataset, method, and experiment. *arXiv preprint arXiv:1705.07632*, 2017.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55. Springer, 2024. doi: 10.1007/978-3-031-72970-6_3.

João Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire de Sousa. Ensemble approaches for regression: A survey. *ACM Computing Surveys*, 45(1):10:1–10:40, 2012. doi: 10.1145/2379776.2379786.

Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. A survey on machine learning for data fusion. *Information Fusion*, 57:115–129, 2020. doi: 10.1016/j.inffus.2019.12.001.

Yair Mundlak. On the pooling of time series and cross section data. *Econometrica*, 46(1):69–85, 1978. doi: 10.2307/1913646.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(3):2354–2367, 2026. doi: 10.1109/TPAMI.2025.3628473.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 44(3):1623–1637, 2022.

Niall Rooney, David Patterson, Sarab Anand, and Alexey Tsymbal. Dynamic integration of regression models. In *Multiple Classifier Systems (MCS)*, volume 3077 of *Lecture Notes in Computer Science*, pp. 164–173. Springer, 2004. doi: 10.1007/978-3-540-25966-4_16.

Dymitr Ruta and Bogdan Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1): 63–81, 2005. doi: 10.1016/j.inffus.2004.04.008.

Yuita Arum Sari and Akio Gofuku. Measuring food volume from RGB-depth image with point cloud conversion method using geometrical approach and robust ellipsoid fitting algorithm. *Journal of Food Engineering*, 358:111656, 2023. doi: 10.1016/j.jfoodeng.2023.111656.

Zhimin Shao, Weibei Dou, and Yu Pan. Dual-level deep evidential fusion: Integrating multimodal information for enhanced reliable decision-making in deep learning. *Information Fusion*, 103: 102113, 2024. doi: 10.1016/j.inffus.2023.102113.

Catherine A. Shipp and Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, 2002. doi: 10.1016/S1566-2535(02)00051-9.

Alexander Smirnov and Tatiana Levashova. Knowledge fusion patterns: A survey. *Information Fusion*, 52:31–40, 2019. doi: 10.1016/j.inffus.2018.11.007.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Chi-en Amy Tai, Matthew Keller, Saeejith Nair, Yuhao Chen, Yifan Wu, Olivia Markham, Krish Parmar, Pengcheng Xi, Heather H. Keller, Sharon Kirkpatrick, and Alexander Wong. NutritionVerse: Empirical study of various dietary intake estimation approaches. In *Proc. 8th Int. Workshop on Multimedia Assisted Dietary Management (MADiMa), ACM Multimedia*, pp. 11–19, 2023. doi: 10.1145/3607828.3617799.

Hikaru Tanabe and Keiji Yanai. CalorieVoL: Integrating volumetric context into multimodal large language models for image-based calorie estimation. In *Proc. 31st Int. Conf. MultiMedia Modeling (MMM)*, volume 15523 of *Lecture Notes in Computer Science*. Springer, 2025. doi: 10.1007/978-981-96-2071-5_26.

Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3405–3414, 2019.

Xiaoqiang Teng, Weiqing Min, Caixia Liu, and Haisheng Li. Multimodal information fusion for food nutrition estimation: A survey. *Information Fusion*, 137:104621, 2027. doi: 10.1016/j.inffus.2026.104621.

Quin Thames, Arjun Karpur, Wade Norris, Fangting Xia, Liviu Panait, Tobias Weyand, and Jack Sim. Nutrition5k: Towards automatic nutritional understanding of generic food. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 8903–8911, 2021.

Michael J. Todd and E. Alper Yıldırım. On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, pp. 1521–1528, 2011.

Gautham Vinod, Jiangpeng He, Zeman Shao, and Fengqing Zhu. Food portion estimation via 3D object scaling. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR) Workshops*, pp. 3741–3749, 2024.

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2642–2651, 2019.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. doi: 10.48550/arXiv.2409.12191.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2507.02546.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b. doi: 10.48550/arXiv.2508.18265.

Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (4):405–410, 1997. doi: 10.1109/34.588027.

Fuyuan Xiao. Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. *Information Fusion*, 46:23–32, 2019. doi: 10.1016/j.inffus.2018.04.003.

Dongjian Yu, Weiqing Min, Qian Jiang, Xing Lin, Xin Jin, and Shuqiang Jiang. OmniFood8K: Single-image nutrition estimation via hierarchical frequency-aligned fusion. *arXiv preprint arXiv:2604.12356*, 2026.

Qingyang Zhang, Yake Wei, Zongbo Han, Huazhu Fu, Xi Peng, Qinghua Hu, Cheng Deng, Cai Xu, Jie Wen, Di Hu, and Changqing Zhang. Multimodal fusion on low-quality data: A comprehensive survey. *Information Fusion*, 135:104437, 2026. doi: 10.1016/j.inffus.2026.104437.

Yudong Zhang, Lijia Deng, Hengde Zhu, Wei Wang, Zeyu Ren, Qinghua Zhou, Siyuan Lu, Shiqing Sun, Ziquan Zhu, Juan Manuel Gorriz, and Shuihua Wang. Deep learning in food category recognition. *Information Fusion*, 98:101859, 2023. doi: 10.1016/j.inffus.2023.101859.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024. doi: 10.26599/AIR.2024.9150038.

A CLAIMS AND EVIDENCE LEVELS

Table 3 lists every claim of the paper with the datasets it rests on, its evidence level and the section that supports it. The appendix then follows the paper: related work (App. B), method details (App. C), datasets, implementation and audits (App. D), replication and protocol variance (App. E), the semantic channel (App. F), the vision–language panel (App. G), decision-level fusion (App. H), class-anchored fusion and its registered tests (App. I), and the extended discussion (App. J).

Table 3: **The claims, and the evidence behind each.**

Claim	Datasets	Evidence level	Key number	Sec.
A prior with no image geometry beats the chain	6 volume	primary on ECUSTFD (ARE; percentile interval, the CR1 interval spans zero; 64/70 targets); log error, a declared sensitivity analysis, on all three family members; exploratory on YCB; directions on MRGBD, NOCS	23/24 cells (24/24 log); pooled log $+0.45$ (t interval $+0.15$ to $+0.75$)	5.2
Geometry adds to the prior when fused within class	ECUSTFD	exploratory; within-class permutation, partition and constant controls; two of four backbones survive Holm	log error from 0.154 to 0.138; beats every nesting combiner	5.5
The same rule holds on samples withheld from every analysis	ECUSTFD held-out, Nutrition5k held-out	**confirmatory**: registered before the held-out runs	$p = 0.016$ (81 objects), $p = 10^{-5}$ (1,738 dishes)	5.6
Fusing several sources within class, not just one	64 pools; ECUSTFD held-out top views	registered and **not confirmed**; exploratory elsewhere	12 pools ahead of 5; held-out $p = 0.64$	5.6
The model predicts the fused gain, and the condition $r^2_{\mathrm{wc}} > 1/n_{\mathrm{eff}}$ flags harm	6 datasets, 82 cells	descriptive; the prediction is a training-fold fit carried out of fold	$r = 0.99$ on mass datasets, 0.45 on volume; 13 of 14 harms flagged	5.5
Untrained fusion gains are mostly shrinkage	ECUSTFD (where testable)	non-inferiority at a post hoc margin, ratio bound	constant/VLM 0.91, bound 0.98	5.4
Trained combiners gain in log space, beyond the constant they contain	6 volume, 46 pools	descriptive (six datasets)	log stack -46%; -28% against a log constant	5.4
The scale stage lacks calibration, not cue combination	MRGBD, YCB, NOCS	exploratory; the factor is fitted on training targets	learned cues match the best one-factor cue	I.5
VLMs beat the class prior mainly where classes vary	5 with a defined bar	exploratory; design confounded	8/66 cases, all on mass datasets	5.3
Rows are not independent evidence	6 volume	descriptive, specific to one contrast	13,371 rows $\approx$ 46 effective	5.1
Humans with the label rival the best pipelines	MetaFood3D, 19 scenes	reference only	crowd 0.623 vs 0.667	F.16

B EXTENDED RELATED WORK

This appendix provides the full related-work discussion the paper condenses, including its positioning table, and then reviews each literature stream it covers.

B.1 THE GAP

Table 4: **The gap this benchmark closes.** Rows are literatures bearing on monocular volume estimation; columns are practices that must be in place before a gain can be attributed to a channel. ✓ standard, • present in some members, × absent, "–" not applicable. The last column asks for per-item records; the last row is the conjunction of the columns, not a claim of novelty in any one: each column is standard practice somewhere.

Literature	Representative work	Metric volume or mass	Category-only baseline	Folds grouped by target	Uncert. clustered on target	Rules vs. per-item oracle	Benchmark validity tested	Per-item records released
Single-view 3D: reconstruction or recognition	(Tatarchenko et al., 2019)	×	✓	×	×	–	•	✓
Shortcuts and blind baselines	(Geirhos et al., 2020)	–	✓	–	×	–	✓	•
Geometric food-volume pipelines	(Dehais et al., 2017; Ando et al., 2019)	✓	×	•	•	×	×	×
Learned monocular food volume	(AlMughrabi et al., 2025; Haroon et al., 2025; Gu et al., 2026)	✓	×	×	×	×	×	•
Food-volume datasets and benchmarks	(Liang & Li, 2017; Abdur Rahman et al., 2023; Sari & Gofuku, 2023; Chen et al., 2024; Thames et al., 2021; Tai et al., 2023)	✓	×	•	×	–	×	✓
Monocular depth benchmarking	(Ranftl et al., 2022)	×	–	×	×	×	•	✓
Food segmentation benchmarking	(AlMughrabi et al., 2026; Alahmari et al., 2024)	×	–	✓	✓	×	•	✓
Leakage and grouped validation	(Kapoor & Narayanan, 2023)	–	–	✓	•	–	✓	•
Benchmark validity and shift	(Torralba & Efros, 2011)	×	–	✓	•	×	✓	✓
Large models as a fusion source	(Han et al., 2026)	•	×	×	×	×	×	×
Decision-level fusion and selection	(Ruta & Gabrys, 2005; Cruz et al., 2018; Shipp & Kuncheva, 2002; Woods et al., 1997; Ko et al., 2008)	–	–	–	–	✓	×	•
Evidential and conflict-aware fusion	(Shao et al., 2024)	–	–	–	–	•	×	×
Forecast combination and stacking	(Bates & Granger, 1969; Granger & Ramanathan, 1984; Breiman, 1996; Claeskens et al., 2016)	–	–	–	–	✓	×	–
Multimodal food-nutrition fusion	(Teng et al., 2027)	✓	•	×	×	×	×	×
OneVol (this paper)	Secs. 3–6	✓	✓	✓	✓	✓	✓	✓

Table 4 sets these literatures against the practices a channel attribution needs. The closest precedent is in single-view 3D reconstruction, where networks were shown to act largely as recognisers: a retrieval baseline matched them (Tatarchenko et al., 2019). The same diagnosis appears as blind or partial-input baselines and shortcut learning in other fields (Geirhos et al., 2020). We carry it

to a metric task, where it can be quantified in physical units. Food-volume work owns the task (Dehais et al., 2017; Ando et al., 2019; AlMughrabi et al., 2025; Haroon et al., 2025; Gu et al., 2026) and the datasets (Liang & Li, 2017; Abdur Rahman et al., 2023; Sari & Gofuku, 2023; Chen et al., 2024; Thames et al., 2021; Tai et al., 2023), while monocular depth (Ranftl et al., 2022) and food segmentation (AlMughrabi et al., 2026; Alahmari et al., 2024) are benchmarked on their own outputs rather than on the volume they feed. Food-volume work compares against other pipelines or a dataset-wide mean rather than against what the object's name implies, and it groups folds by scene (Tai et al., 2023) or by unseen instance (Ando et al., 2019) without clustering intervals on the objects. Grouped validation and leakage are well understood elsewhere (Kapoor & Narayanan, 2023); what remains unmeasured is what ignoring them costs here. Benchmark-validity work asks whether a dataset supports a ranking (Torralba & Efros, 2011), not whether it supports a measurement. VLMs are proposed as fusion sources because of their generality (Han et al., 2026), not because they beat the trivial alternative of naming the object.

B.2 Position within information fusion

The task is a fusion problem before applying any rule. An estimate already combines a recogniser's identity, a depth model's shape, a calibration step's scale and an integration operator's assumption about the unseen half of the object. The combination we study sits at the decision level in its most tractable form: every source emits one scalar in the same physical unit, so the classical apparatus applies as written, from untrained and robust rules and dynamic selection to conflict handling and the per-sample oracle (Ruta & Gabrys, 2005; Cruz et al., 2018; Shipp & Kuncheva, 2002; Woods et al., 1997; Ko et al., 2008; Shao et al., 2024). A survey of multimodal nutrition estimation names portion ambiguity and within-class variability as open problems (Teng et al., 2027), the multisensor and machine-learning fusion surveys supply the level-and-reliability taxonomy we use (Khaleghi et al., 2013; Meng et al., 2020), and our conflict protocol asks when a degraded source makes fusion worse (Zhang et al., 2026). Two results from that literature shape our reading. Simple averages often beat estimated weights, the forecast-combination puzzle (Bates & Granger, 1969; Claeskens et al., 2016), while regression combinations and stacking fit the weights and an intercept to held-out truth (Granger & Ramanathan, 1984; Breiman, 1996). And for a mean combiner, the ensemble's squared error equals the members' mean error minus their spread, the ambiguity decomposition (Krogh & Vedelsby, 1995). Both explain why pooling variants of one chain recover little. The task adds ground truth in physical units, so a claimed gain can be audited rather than only ranked. We combine at the decision level because every source reports in the same unit; we compare earlier fusion at the depth-field level in App. H.9.

B.3 Multimodal fusion for food and object measurement

The fusion literature organises multi-source estimation by the level at which sources are combined (data, feature or decision) and by how source reliability and conflict are handled (Khaleghi et al., 2013; Meng et al., 2020). Within food computing, a recent survey organises nutrition estimation around visual–geometric multimodal learning, semantic and knowledge-augmented estimation, and compositional multimodal reasoning, and identifies portion ambiguity and intra-class variability as open problems (Teng et al., 2027); OneVol is a decision-level study of exactly those two problems. Category recognition itself is a mature fusion problem in food computing (Zhang et al., 2023), and knowledge-driven fusion, which injects a prior in place of a measurement, has its own pattern language (Smirnov & Levashova, 2019); the category prior of Sec. 3.1 is the simplest instance of that pattern. Early systems relied on stereo, RGB-D sensing or explicit geometric fitting (Dehais et al., 2017; Ando et al., 2019; Graikos et al., 2020), while dataset papers established the breadth of sensing regimes now in use (Liang & Li, 2017; Abdur Rahman et al., 2023; Sari & Gofuku, 2023; Chen et al., 2024; Thames et al., 2021; Tai et al., 2023).

Of these, we evaluate all but NutritionVerse, whose plated scenes fall within the within-class-varying case Sec. 5.3 identifies as the informative one and which is the first dataset a follow-up should add. Its ground truth is synthesised per scene rather than measured per object, which is why it is absent here and not a judgement on the dataset. Learned monocular pipelines have since become competitive (AlMughrabi et al., 2025; Haroon et al., 2025; Gu et al., 2026); the accounting this paper asks for is one they do not provide. A non-visual control is not foreign to this literature: portion-estimation work routinely reports a baseline that predicts the dataset mean for every instance and beats it comfortably (Thames et al., 2021; Vinod et al., 2024). That control is *global*, a single scalar

for the whole dataset. What Sec. 3.1 adds is its class-conditional refinement, a per-category median fitted on training folds only, and the observation that this version is not beaten by the full monocular chain on five of six volume datasets. Support estimation remains a major source of downstream variation (Kirillov et al., 2023; Alahmari et al., 2024). OneVol fits within that taxonomy but inverts the usual direction: rather than proposing a fusion architecture, it measures how much each channel contributes and what fusing them can achieve on the datasets the field already uses.

B.4 Monocular and metric depth

Monocular depth has evolved from supervised regression (Eigen et al., 2014) to self-supervised photometric training (Godard et al., 2019) and, more recently, to foundation-scale and metric-aware models (Ranftl et al., 2022; Bhat et al., 2023; Piccinelli et al., 2026; Bochkovskii et al., 2025; Wang et al., 2025a). These are typically evaluated on depth-centric benchmarks (Koch et al., 2020). The downstream question here is what happens when a depth field is inserted into a metric pipeline with its own support, scale, and integration decisions. Our integration operators follow standard volumetric and minimum-volume formulations (Curless & Levoy, 1996; Todd & Yıldırım, 2007).

B.5 Benchmark validity

Evaluation-focused studies have repeatedly shown that metric choice and cross-dataset transfer reorder model rankings (Ranftl et al., 2022; Koch et al., 2020), and that dataset bias changes which model appears strongest (Torralba & Efros, 2011). Frameworks such as WILDS formalise reporting across environments (Koh et al., 2021), and the fusion literature has begun to ask when combining sources helps and when a degraded or conflicting source makes it worse (Zhang et al., 2026). Our validity criteria extend that concern one level further: before asking whether a ranking transfers, we ask whether the dataset can measure the quantity.

B.6 Fusion rules, conflict and selective prediction

Decision-level fusion of continuous estimates relies on a small set of untrained rules (averaging, robust order statistics, precision weighting) and per-sample *selection*, in which a competence estimate selects which source to trust for each input (Ruta & Gabrys, 2005; Cruz et al., 2018). Selection is only worth learning when sources fail on different inputs, which is why source diversity is the classical precondition for fusion gain and why the per-sample oracle, standard inside dynamic selection (Ruta & Gabrys, 2005; Cruz et al., 2018), is the natural ceiling against which a rule should be reported. Conflict between sources is handled in the evidential tradition by measuring disagreement and redistributing mass away from conflicting evidence, applying robust combination rules designed to degrade gracefully as conflict increases (Lefevre et al., 2002; Florea et al., 2009; Xiao, 2019); our conflict-aware rule is a scalar analogue, and the conflict-injection protocol of Sec. H.11 serves as its stress test. Simple untrained rules are often as good as learned combiners, and their advantage tracks source diversity (Shipp & Kuncheva, 2002); for continuous estimates, this derives from decomposing an average's error into the members' mean error minus their spread around the average, so that a second source pays in proportion to how differently it fails, and further sources add less. We reproduce both the empirical finding and this level of saturation. Dynamic selection methods (Ruta & Gabrys, 2005; Cruz et al., 2018) learn a competence estimate for each input; we train a random-forest selector of that kind, score three published schemes beside it (Sec. H.3.1), and the oracle ceiling bounds all such methods. Evidential deep fusion has recently revisited the same questions using learned, uncertainty-aware combiners (Shao et al., 2024). Calibrated uncertainty (Kendall & Gal, 2017; Lakshminarayanan et al., 2017) and selective prediction with risk–coverage control (Geifman & El-Yaniv, 2019) provide the machinery for abstention. Large VLMs have recently been incorporated into fusion systems as another source, typically as a semantic or reasoning channel alongside metric sensors (Han et al., 2026); within food computing, they are already known to recover volume poorly, which propagates into their calorie estimates (Tanabe & Yanai, 2025). We treat one as exactly that source and measure its contribution. What this literature lacks, and what OneVol supplies, is a test to determine whether the benchmark used to score the fusion can distinguish source contributions.

C METHOD DETAILS

This appendix fixes the vocabulary and gives the parts of the method the paper states only briefly: the fusion rules, the measurement-model estimation details, the headroom metric, the validity criteria, what we report and why, and the sensitivity of the thresholds.

C.1 VOCABULARY

Table 5 fixes the vocabulary used throughout.

Table 5: **Terms, used in one sense throughout.**

Term	Meaning
target	one physical object; the independent unit and the bootstrap cluster
item	one evaluated row: an object in one image
source	one configuration of Eq. (1), a prior, a VLM run or a constant
pool, member	a set of sources fused together, and each source in it
class prior	median volume of same-class training targets (semantic channel)
global prior	median volume of all training targets
best learnable prior	the class prior where two classes hold several targets, else the global prior
reference chain	each dataset's configuration fixed before the study
incumbent VLM	Qwen2.5-VL-7B, the model fixed before the panel was assembled
panel model	one of the fourteen VLM members of Sec. 5.3
selector	a learned per-item choice between sources
cell	one (dataset, source) or (dataset, backbone) combination
ARE, log error	$\lvert\hat{V} - V\rvert/V$ and $\lvert \log \hat{V} - \log V\rvert$, averaged over items (Sec. 3.3)
oracle, headroom H	the best source per item in hindsight, and the share of its gain a rule captures
σ_w	within-class log spread of the truth: the floor on the class prior's log RMSE
r_{wc}, n_{eff}	a source's within-class correlation with the truth, and the targets that inform its slope
design effect	the variance inflation of rows clustered in targets (Kish)
direction	the sign of a comparison on a dataset that fails TARGETS
§	the dataset fails TARGETS: only the sign is reported

C.2 FUSION RULES IN FULL

Six untrained rules are scored: the mean, the median, the trimmed mean, the geometric mean, a dispersion-weighted mean, and a conflict-aware rule. The dispersion-weighted mean weights each source by the inverse variance of its *predictions* across the pool's items. It needs no ground truth and is not a reliability weight: on a pool with a near-constant member, it collapses onto that member. The conflict-aware rule averages while the pool's relative spread, $(\max_s \hat{V}^{(s)} - \min_s \hat{V}^{(s)})/\operatorname{median}_s \hat{V}^{(s)}$, stays below $\tau = 0.5$, and takes the median above it. On a two-source pool, which 28 of the 46 reported pools are, the six collapse to three estimators. A learned *selector*, a random forest on pre-integration cues (in-mask depth statistics, mask area and compactness, segmentation-quality proxies), predicts which source to trust per item, in a classification and a regression variant. We select the rule for a pool by nested selection: we score the untrained rules and selectors on the training folds, apply the winner to the held-out fold, and never choose a rule on the rows it is scored on. On the same folds we also score trained combiners: a precision-weighted mean, least-squares stacks fitted in cubic centimetres and in log space (Granger & Ramanathan, 1984; Breiman, 1996), a selector that reads the sources' own outputs, and three published dynamic-selection schemes, overall local accuracy (OLA) (Woods et al., 1997) and the k-nearest-oracles rules KNORA-E and KNORA-U (Ko et al., 2008) (App. H.3.1). We fit and score every combiner and rule out-of-fold (OOF). The class-anchored rule follows from the measurement model.

C.3 THE MEASUREMENT MODEL: ESTIMATION DETAILS

The accuracy bar. Out of fold, a class centre estimated from n_c targets inflates this by about $\sqrt{1 + 1/n_c}$. We estimate σ_w by pooling within-class variances over distinct targets, each class weighted by its degrees of freedom. Sec. 5.1 reports it per dataset.

Fitting the class-anchored rule. We fit $\hat{m}_c$ as the class median and b by least squares within class on the training folds, clip d at three robust standard deviations so that one degenerate output cannot move the slope, and, where a test item's class is absent from the training folds, anchor on the global median with the source's deviation from its global mean. Eq. (3) is the fusion counterpart of the within-class variance explained by a channel beyond its category, which Sec. 5.5 measures. The fit minimises squared log error, so the exponential of the fused estimate is a median, not the point that minimises ARE.

The prediction and the fusion condition. Two caveats bound that prediction. It is the training-fold fit of the same regression carried out out-of-fold, so it tests how well the fit transfers, not Eq. (2) itself. And estimating b from n_{eff} independent targets costs about $\sigma_w^2/n_{\text{eff}}$, so the rule should help only where Eq. (4) holds, counting the targets that inform the slope: those in classes with at least two, or all of them where the anchor is global. We call Eq. (4) the *fusion condition*. It needs no fusion to evaluate, only a source's within-class correlation on training folds and a count of independent targets, so a practitioner can decide whether to fuse at all before writing a combiner: a catalogue of 20 product weights beside a load cell needs a within-class correlation above 0.22 before a fitted slope is worth its variance, whatever the sensor costs.

Replication. A paired difference between two channels, observed on n_t targets whose rows correlate at ρ within target, has a variance inflated by Kish's design effect, $1 + (m_w - 1)\rho$ with $m_w = \sum_t m_t^2 / \sum_t m_t$ for unequal clusters (Kish, 1965). Rows beyond a handful per target add almost nothing, and the number of targets needed to detect a difference Δ at 80% power is about $(z_{0.975} + z_{0.8})^2 s_t^2/\Delta^2$, where s_t is the spread of per-target mean differences.

C.4 Ceilings and the headroom metric

We score two errors: mean ARE, $A = \frac{1}{n}\sum_i |\hat{V}_i - V_i|/V_i$, and the mean absolute log error, $L = \frac{1}{n}\sum_i |\log \hat{V}_i - \log V_i|$, the *symmetric log error*, with non-positive outputs floored at $1\,\text{cm}^3$; the log RMSE is the root mean square of the same residual. For a pool with ground truth V, the oracle *selection* estimate is $\hat{V}_i^{\text{orc}} = \hat{V}_i^{(s_i^\star)}$ with $s_i^\star = \arg\min_s |\hat{V}_i^{(s)} - V_i|/V_i$. The headroom a rule Φ captures is

$$H(\Phi) = \frac{A(\text{best single}) - A(\Phi)}{A(\text{best single}) - A(\hat{V}^{\text{orc}})}. \tag{5}$$

$H \le 0$ means the rule does not beat choosing the best single source, and $H > 1$ is possible for a blend, since the oracle bounds selection, not combination. We report H with a target-clustered interval and withhold it when the gap's interval includes zero. A rule *improves* on a pool only when its error falls at least 1% below the best member's, in relative terms.

C.5 Validity criteria: what each one licenses

A dataset must measure volume error before any of the above means anything. We define five criteria. VARIATION: the target varies across items. TARGETS: at least 30 distinct physical targets, an object counted once however many views it contributes. CLASSES: more than one class. SPREAD: the class does not determine the volume, screened by the within-class coefficient of variation over distinct targets at a nominal 0.10, a free parameter that App. C.7 sweeps; it is applicable only where at least half the classes hold two distinct targets. SKILL: the chain beats the best learnable prior, the cross-validated class or the global median. SKILL describes the pair of benchmark and pipeline rather than the data, so a better chain flips it, and we count the two kinds of failure apart.

Each criterion licenses a different claim. Comparing two sources on the same items needs VARIATION and TARGETS. Reading an error as metric accuracy needs all five. On a dataset that fails TARGETS we still report every comparison, since a point estimate needs no clusters, but its intervals resample too few objects to support a significance claim, so we mark them § and report only the sign, which we call a *direction*. A within-class correlation needs more: at least 50 distinct targets and three classes that hold several of them. The reasons for each threshold are in App. C.7.

C.6 What we report, and why

A benchmark is only as good as the quantity it reports, and the choice is not neutral here, so we state it and its costs.

Absolute relative error

The headline quantity is the mean absolute relative error, $\text{ARE} = |\hat{V} - V|/V$, averaged over items. Two data properties require a relative measure. Target values span two orders of magnitude across datasets and a factor of nine within ECUSTFD alone (Table 1). An error in cm^3 would be dominated

Table 6: **Four published pipelines against the same prior.** Our reimplementations, scored on the targets each shares with the reference chain. Last column: the paired difference $\mathrm{ARE}_{\text{method}} - \mathrm{ARE}_{\text{prior}}$ (instance-clustered 95% interval), positive where the prior is more accurate. Bold, the interval excludes zero on a dataset passing TARGETS; §, it fails, and below ten joined targets the interval is withheld; †, the run returns volumes three orders of magnitude too small and carries no verdict. Mean ARE caps an under-estimate at 1.0, so the median ratio and the log error beside it separate a measurement from a unit failure. Intervals are nominal: this table corroborates Sec. 5.2 rather than joining its confirmatory family, and bounds our reimplementations, not the original methods.

Method	n	Method ARE↓	Med. ARE↓	Med. bias	Log err.↓	Chain ARE↓	Prior ARE↓	Paired diff.	95% CI
ECUSTFD, 62–70 distinct targets									
ECUSTFD coin-reference	771	0.432	0.270	×1.25	0.38	7.010	0.154	**+0.277**	**[+0.17, +0.40]**
MADIMA23 pipeline (RealSense)	837	0.515	0.544	×0.47	0.95	6.493	0.159	**+0.356**	**[+0.28, +0.43]**
DepthCalorieCam	837	0.872	0.899	×0.10	2.31	6.493	0.159	**+0.713**	**[+0.65, +0.76]**
MADIMA23 pipeline (ZoeDepth)†	837	0.995	0.996	×0.01	4.86	6.493	0.159	+0.836	[+0.78, +0.88]
MRGBD point-cloud†	837	1.000	1.000	×0.01	4.93	6.493	0.159	+0.841	[+0.79, +0.88]
MRGBD§, 8 distinct targets joined									
MRGBD point-cloud	21	0.422	0.425	×0.58	0.56	0.343	0.128	+0.294	–
MetaFood3D, 34 distinct targets									
MADIMA23 pipeline (ZoeDepth)	34	0.750	0.323	×1.00	0.56	1.597	1.673	**-0.923**	**[-2.00, -0.06]**
MADIMA23 pipeline (RealSense)	34	0.804	0.509	×0.81	0.91	1.597	1.673	**-0.869**	**[-1.80, -0.10]**
MRGBD point-cloud	34	0.846	0.386	×0.97	0.58	1.597	1.673	-0.827	[-1.89, +0.01]
DepthCalorieCam†	34	0.999	0.999	×0.01	4.38	1.597	1.673	-0.674	[-1.88, +0.24]
NOCS§, 15 distinct targets									
DepthCalorieCam†	13,371	0.999	0.999	×0.00	5.88	3.287	0.521	+0.478	[+0.15, +0.73]
MADIMA23 pipeline (ZoeDepth)†	13,371	0.999	1.000	×0.00	5.88	3.287	0.521	+0.478	[+0.15, +0.73]
MRGBD point-cloud†	13,364	1.291	1.000	×0.00	5.86	3.285	0.521	+0.770	[+0.33, +1.21]

by the largest items and would not be comparable between datasets. The food-volume literature reports relative error, so we can compare our numbers with published results. ARE is also the quantity a dietary application cares about, since energy varies with volume.

What ARE hides, and what we add

ARE is asymmetric: an estimate that is half the truth scores 0.5, while one that is twice the truth scores 1.0 and one that is ten times the truth scores 9.0. Thus, a method that over-estimates is punished more than one that under-estimates by the same factor, and a single gross over-estimate can move a mean. This is not a technicality in this domain: one backbone returns $1{,}925\,\mathrm{cm}^3$ for a $373\,\mathrm{cm}^3$ apple (Fig. 1). We therefore report the *median* ARE, which ignores that tail, wherever a mean is tail-driven (Sec. 5.2) and, with the *outlier rate*, the fraction of items with $\mathrm{ARE} > 1$, that is, worse than predicting zero, in the released per-run tables. Where the question is how much a channel knows rather than how large its error is, we use ΔR^2 in log-volume (Sec. 5.5), which treats a factor of two in either direction alike. Reporting the mean alone would flatter robust rules, and reporting the median alone would hide the gross failures fusion is supposed to repair, so we report both.

Coverage

A source that returns no estimate, either because its depth is invalid within the mask or because a VLM member fails to emit a number, is not scored on that item. An error computed over a subset is not comparable with one computed over the whole set, so we report coverage alongside every error and align each pool to the items answered by all of its sources.

Uncertainty and effective sample size

Repeated views of a single physical target are not independent observations, so we count the independent units, distinct targets, rather than rows, and cluster each bootstrap interval by target instance; the effective sample size under clustering lies between the two counts and is bounded by the measured interval inflation of Sec. 5.1. Every interval is a percentile interval of that clustered bootstrap. Sec. 5.1 shows the cost of ignoring this.

SKILL AGAINST A PRIOR

An ARE alone cannot indicate whether a system measured anything, because the number that would follow from naming the object is unknown to the reader. Every error is therefore accompanied by the trivial global and category priors, and by the skill score $S = 1 - \mathrm{ARE}_{\text{method}}/\mathrm{ARE}_{\text{prior}}$, which is negative when the prior wins.

FUSION-SPECIFIC QUANTITIES

For a pool, we add the oracle ceiling and the captured headroom H of Eq. (5), since a fusion result without a ceiling cannot be distinguished from one with no headroom left, and the pairwise correlation of signed relative error, which the fusion literature says a gain should depend on.

MULTIPLICITY AND WHICH CLAIMS ARE CONFIRMATORY

Table 49 reports an interval for each of 46 pools, and App. H.4 adds a correlation across them. These are nominal 95% instance-clustered intervals with no family-wise correction, so at the nominal level roughly two of the 46 would be expected to exclude zero by chance alone. The count of 20 pools improving on their best member is itself a selection over the same family. We therefore treat every per-pool verdict, the diversity analysis, and the cue-group ablation as *exploratory*: they describe where headroom lies, and no claim in this paper rests on any single one of them. One result is *confirmatory* and is the one the paper's argument uses: the paired comparison of the chain against its best learnable prior on the family of the three volume datasets that passed TARGETS when the rule was fixed (Sec. 5.2), corrected at a Bonferroni level of $0.05/3$. We fixed the membership rule and level before running the paired analysis, but we deposited no dated pre-registration, and the reader should weigh the family accordingly. YCB passed TARGETS only after its rebuild (Sec. 5.1); rather than letting a member added after the fact change the level applied to the pre-specified three, we report its comparison as exploratory wherever it appears. The comparison against the reimplemented pipelines (Table 6) is corroborative and carries nominal intervals, and the effect of fold grouping (Sec. E.5) is a protocol demonstration reported as ratios without a test. The aggregate statements drawn from the exploratory family–that untrained rules remain competitive with the learned selector, and that gains do not grow past two sources–are counts over the whole family rather than claims about any member, and are unaffected.

WHAT WE DO NOT REPORT

We do not report IoU or Chamfer distance against a reconstructed shape: only MetaFood3D ships a mesh aligned with the capture, so a shape metric would be defined on a single dataset of thirty-four items. We do not report calorie error as a headline, because it is a deterministic function of volume under a fixed density and would merely restate the same result in different units.

C.7 SENSITIVITY OF THE VALIDITY THRESHOLDS

Three numbers are ours to choose: TARGETS at 30 distinct targets, SPREAD at a within-class coefficient of variation of 0.10, and the 50 targets over three multi-target classes that App. C.5 requires before a *within-class correlation* is reported. None is derived from first principles, so Table 7 reports which verdicts move across a plausible range of each. Between 20 and 30 targets, nothing moves. At 50 three more datasets fail TARGETS, namely MADIMA23 with 36 distinct targets, YCB with 40 and MetaFood3D with 34; MRGBD (9) and NOCS (15) fail at every threshold, and ECUSTFD (70) passes at every one. SPREAD is stable between 0.10 and 0.15 and moves at 0.05, where ECUSTFD (0.098) and MRGBD (0.053) both pass. The two verdicts a looser threshold would overturn are therefore those SPREAD failures, and neither changes what the dataset may be used for: ECUSTFD also fails SKILL, and MRGBD also fails TARGETS.

The TARGETS criterion now governs two things at once, since the same count determines both whether an absolute error is treated as accuracy and whether an instance-clustered interval has enough units to resample; so the sweep has to be read against the confirmatory family as well. It does not move it. At 20, the family is unchanged, because NOCS at 15 and MRGBD at 9 fail anyway. At 50, it shrinks to ECUSTFD alone, dropping MADIMA23 and MetaFood3D. ECUSTFD's difference then needs no multiplicity correction and excludes zero by more than it does under the

family of three. The confirmatory conclusion of Sec. 5.2 therefore holds unchanged for any TARGETS threshold between 20 and 50, which is the whole range we would defend.

The correlation threshold is a number set by observation rather than judgement, and we explain why. A correlation is defined *inside* a class, so it needs classes that each hold several targets, not merely a sufficient total; at MADIMA23's 34 targets over 32 classes, only two of which hold more than one target, the clustered bootstrap returns $[0.000, 0.843]$, which is a collapse rather than an interval. We therefore set the requirement above the point where that happens rather than at the TARGETS floor of 30, and report no correlation for any dataset below it. The verdicts that decide admissibility do not depend on where a line is drawn; where a verdict does depend on it, as this one does, we report the line with the evidence that fixed it.

Table 7: **Sensitivity of the two numeric criteria.** TARGETS (the distinct-target count) and SPREAD (within-class CV) at three thresholds each; ✓ passes, "–" marks SPREAD inapplicable. Sec. C.7 reads the sweep.

		TARGETS at				SPREAD at		
Dataset	**Dist. targets**	20	30	50	**Within CV**	0.05	0.10	0.15
ECUSTFD	70	✓	✓	✓	0.098	✓	×	×
MADIMA23	36	✓	✓	×	0.073	–	–	–
MRGBD	9	×	×	×	0.053	✓	×	×
MetaFood3D	34	✓	✓	×	–	–	–	–
YCB	40	✓	✓	×	–	–	–	–
NOCS	15	×	×	×	0.332	✓	✓	✓

D DATASETS, IMPLEMENTATION AND AUDITS

This appendix describes the datasets, the geometric sources and the metric scale in full, and then the reference implementation, the ground-truth audits, the scale protocol, the checks of the two upstream stages against the truth their datasets ship, and the compute.

D.1 DATASET DETAILS

We evaluate on ECUSTFD (Liang & Li, 2017), MADIMA23 (Abdur Rahman et al., 2023), MRGBD (Sari & Gofuku, 2023), and MetaFood3D (Chen et al., 2024) in its challenge release of 34 scanned items with one evaluation image each, a YCB split (Calli et al., 2015) and a NOCS split (Wang et al., 2019). For the VLM panel, we add Nutrition5k (Thames et al., 2021) and OmniFood8K (Yu et al., 2026), whose ground truth is mass in grams; they enter through the semantic channel and are never pooled with the volume datasets. The datasets fall into three groups: single foods photographed in isolation (ECUSTFD, MRGBD), plated meals (MADIMA23, MetaFood3D, Nutrition5k, OmniFood8K) and household objects (YCB, NOCS-REAL275, shortened to NOCS). Table 1 gives what each contains and what it can support, and Fig. 3 shows its images.

A *target* is one physical object and an *item* is one evaluated row; this distinction matters more than any modelling choice in the paper. Rows greatly outnumber targets in four datasets: ECUSTFD contributes 12 views per target, MRGBD 16, YCB 20 and NOCS 891. A class constrains the target tightly on the food volume benchmarks, with within-class coefficients of variation of 0.05–0.10, and less so on NOCS (0.33) and on the two mass datasets (0.47 and 0.27), which we added because their portions vary within a class.

We take targets as each dataset distributes them, except for YCB and NOCS, which ship meshes whose volumes we integrate. Two audits changed the benchmark, and App. D.5 gives both in full. On NOCS, no mesh is watertight, and the three laptop models fill 0.6–5% of their own convex hull, which no rigid object does; we exclude the laptop class, leaving 15 targets. On YCB, registered depth existed for one of forty objects; registering the infrared depth into the RGB frame rebuilds the split to 800 items over all 40. Everything else is used as distributed, so an annotation defect is a defect in our numbers too.

Every chain and prior is scored on the evaluation split each dataset defines; on ECUSTFD that is the test split, 837 side-view items over 70 targets. The panel scores every released crop (2,867 over 143 targets), and cross-channel comparisons join on the 803 items both cover.

Full notes to Table 1. Targets counts distinct physical objects and *Views* is rows per target; the within-class CV is the median over multi-target classes, with that count in parentheses. Ranges and medians in cm^3; † mass in grams. *Fails*: VARIATION, the target varies; TARGETS, at least 30 distinct targets; CLASSES, more than one; SPREAD, within-class CV ≥ 0.10, applicable only where at least half the classes hold two targets; SKILL, the chain beats the best trivial prior, which describes the pipeline rather than the data; ‡, evaluated on the incumbent VLM; *, SKILL passes on mean ARE by a difference whose interval spans zero and fails on the log error. *Licensed use*: comparisons between sources on the same items need VARIATION and TARGETS; accuracy needs all five; a dataset failing TARGETS gives directions. NOCS-REAL275 is shortened to NOCS.

D.2 THE GEOMETRIC SOURCES

Each configuration of Eq. (1) instantiates its stages with published components. The support $\mathcal{S}$ is a GroundingDINO proposal (Liu et al., 2024) prompted with the class label and refined by the Segment Anything Model (Kirillov et al., 2023); on the two object datasets, the support axis adds BiRefNet (Zheng et al., 2024), a class-agnostic salient-object segmenter. The depth $\mathcal{D}$ comes from one of four metric monocular backbones, ZoeDepth (Bhat et al., 2023), Depth Pro (Bochkovskii et al., 2025), UniDepthV2 (Piccinelli et al., 2026) and MoGe-2 (Wang et al., 2025a), or, on MADIMA23's sensor axis, from the dataset's RGB-D capture. The operator $\mathcal{V}$ is the convex hull or minimum-volume ellipsoid of the points (*hull*), a truncated signed-distance integration, a height map over a RANSAC support plane, or an irregularity-weighted blend of hull and height map. Table 46 lists the VLM panel with its references, and Table 45 lists its prompts. The *reference chain* of each dataset is one configuration fixed before the study (Sec. 5.2).

D.3 THE METRIC SCALE

One rule sets λ for every configuration, with no constant fitted to the evaluation target. Each capture contributes the cues it affords: an annotated reference object (the ECUSTFD coin), the dataset's sensor depth (MRGBD, NOCS and YCB), the support plane at a documented capture distance, and a class-agnostic size prior. λ is their confidence-weighted median, with weights fixed in code: reference object and sensor at 1, the plane at its RANSAC inlier ratio, the size prior below any physical cue. When a configuration falls back on the plane's assumed height or the size prior's fixed extent, its volume rests on an assumption rather than on the scene, as in MetaFood3D. On every dataset with a physical scale cue, the four backbones, whose native scales differ by up to a factor of three, recover the same scene depth to within 1.09–$1.31\times$ (Apps. D.6 and D.4). Because the rule is itself a decision-level fusion of cues, Sec. I.5 treats it as one.

D.4 REFERENCE IMPLEMENTATION

Every geometric source is a configuration of the released pipeline. Support $\mathcal{S}$ is a label-guided GroundingDINO (Swin-T) proposal refined by the Segment Anything Model (SAM, ViT-H) for every reference configuration on every dataset; no configuration uses dataset-provided masks, so the support source is identical wherever another axis is being read. Segmentation quality on food imagery varies more between methods than the depth backbones vary among themselves (Alahmari et al., 2024), and the ordering shifts again once the viewpoint moves: across 35 configurations benchmarked on food video, mean average precision spans 29.6 to 95.4 (AlMughrabi et al., 2026), a preprint that is not yet peer reviewed. Because that evidence is unrefereed, the released runs also include the measurement directly, and they present it as a fusion axis rather than a substitution: on NOCS and YCB the sweep includes a BiRefNet support member with backbone, operator, viewpoint, sensor and scale rule held identical, and Sec. 5.4 reports what combining the two supports achieves. The TARGETS criterion determines where that axis may carry an interval. YCB supplies 40 distinct objects, so its support pool is the first segmentation comparison in this paper entitled to one; NOCS supplies 15 and is not. The contrast is visible on NOCS: bootstrapping the NOCS support difference across rows gives -7.45 $[-8.53, -6.39]$, which reads as settled, while resampling its 15 target instances gives the same -7.45 with $[-14.67, -0.97]$, an interval 6.4 times wider. The direction survives here; the row count does not license the interval.

Depth $\mathcal{D}$ is one of four metric monocular backbones: ZoeDepth (Bhat et al., 2023), Depth Pro (Bochkovskii et al., 2025), UniDepthV2 (Piccinelli et al., 2026) and MoGe-2 (Wang et al., 2025a), or, on the sensor axis, the dataset's own RGB-D capture. The sensor axis is opened as a pool

only on MADIMA23, whose two sensors capture the same items; on MRGBD, YCB and NOCS the shipped depth serves as a scale cue and as the reference for the component check of App. D.7 rather than as a second pool member. Back-projection needs intrinsics, and only NOCS provides its own. We give the field of view assumed for each: ECUSTFD, MADIMA23 and MetaFood3D $50^\circ \times 35^\circ$, MRGBD $90^\circ \times 58.7^\circ$, YCB $55^\circ \times 42^\circ$, NOCS $56.9^\circ \times 44.3^\circ$, the last derived from the shipped calibration. A volume scales as the inverse of the focal-length product in its lateral extent, so a five percent error in an assumed focal length moves every volume on that dataset by ten percent, as Sec. G.6 states for Nutrition5k. Two of these deserve flagging rather than defending: YCB ships per-camera intrinsics that our registration already reads and our back-projection does not use, and MRGBD's pair is the depth module's field of view rather than the colour stream's, which would over-project its lateral dimensions. Neither affects the verdict, because both datasets are compared against priors on their own items and a scale error is common to chain and prior alike, but both are implementation errors, and we name them. Metric scale λ is set by a single rule, identical across all configurations, with no constant fitted to the evaluation target anywhere in the sweep. Two of the cues nevertheless carry a fixed prior rather than a scene measurement, and we name them here because the distinction matters for how a chain's error should be read. The plane cue targets the midpoint of the documented camera-height range, and the class-agnostic size cue divides the object's median extent into a fixed 0.08 m; its stated bound of 3–20 cm cannot bind, since 0.08 lies inside it, so that cue does not bound the extent but sets it. Neither constant is fitted to any dataset, nor is it recovered from the image; where a configuration falls back to them, its volume is a shape factor times an assumption. The released scale report records, per item, which cue fixed λ, and the fractions are uneven: on MetaFood3D, which supplies no reference object, no sensor and no documented distance, the reference chain takes the size cue on 33 of 34 items while the other three backbones take the plane cue on 27 to 30. That is the mechanism behind the $3.11\times$ spread those four backbones show there, and a reason to read MetaFood3D's chain numbers, including its SKILL pass, as resting on an assumed object size rather than on a measured scene. Each capture contributes the cues it physically affords, combined by a confidence-weighted median, $\lambda = \mathrm{wmedian}\{(\lambda_c, w_c)\}$, with the weights fixed in code and identical everywhere: the reference object and the sensor enter at $w = 1$, the support plane at its RANSAC inlier ratio, and the class-agnostic size prior at $\min(0.9, k/5)$ for k measured object extents, so that it never outranks a physical cue; a documented rig distance sets the plane's target depth rather than entering as a cue of its own. The cues are: the annotated reference object, where one exists (the ECUSTFD coin, whose apparent diameter under each backbone measures that backbone's own scale error); the dataset's shipped sensor depth, where there is one (MRGBD, NOCS and YCB, as the median ratio between sensor and predicted depth over the frame, with YCB's infrared depth first registered into the RGB frame); the documented capture distance, where the rig records one (MADIMA23's 40 and 60 cm); and, on MetaFood3D alone, nothing stronger than a generic tabletop prior – a support plane between 0.35 and 0.70 m and a class-agnostic object extent set to 0.08 m (above).

The recovered scale is an output of the run, released per image and per cue. A constant fitted inside a cell would absorb exactly the difference its axis is meant to measure, which is why none exists anywhere in the sweep; App. D.6 reads the recovered scales against each dataset's own reference as the check that the rule landed. Pools inherit their comparability from the rule rather than from a shared number: two viewpoints of the same objects legitimately resolve different numeric scales, because the rule is applied to each view's own scene, and the loader refuses a pool whose members were not all produced under this rule. Back-projection $\mathcal{B}$ lifts the in-mask depth to a point cloud, voxel-downsampled at 5 mm with statistical outlier removal. The integrating operators $\mathcal{V}$ are *hull* (*geom* in the released files and tables: convex hull or minimum-volume ellipsoid, chosen per item by mask compactness), *TSDF* (truncated signed-distance integration of the single view), *height-map* (height above a support plane fitted by random sample consensus, RANSAC) and *hybrid* (an irregularity-weighted combination of the hull and height-map estimates); an operator pool contains only operators evaluated with the same backbone, view and sensor as the reference configuration, and each configuration contributes one prediction per item. The pipeline also offers a post hoc multiplicative corrector fitted on a random row split of the same dataset; because that split is not grouped by instance, it is exactly the leak this benchmark rules out, and this paper uses no corrected run.

Priors use 5 grouped folds with a fixed seed; every interval in this paper is a $20{,}000$-draw bootstrap, clustered on the target instance, with the family-wise bounds of Sec. 5.2 set at $\alpha/3$ using the same draws. We use a single draw count throughout rather than tuning it per table: several of these

intervals include a printed decision, and at 2,000 to 3,000 draws, two of them fall on lower bounds inside 0.007 of zero, which is a significance call that moves with the seed rather than with the data. One convention is worth stating in one place, because it differs by measure. A source can emit a non-positive volume, which mean ARE scores at one, a log error cannot take at all, and a log-space regression cannot use. So mean ARE keeps such items, the log error floors them at $1\,\mathrm{cm}^3$ rather than dropping them, and the ΔR^2 of Sec. 5.5 drops them, which is why its n falls short of the dataset (on ECUSTFD, 830 of 837). Dropping the chain's worst outputs makes the channel-information measure kinder to the chain than the error tables are, which is the conservative direction for the claim that the measure supports. The interval itself is a percentile bootstrap on a mean, which is the least assuming choice and not the best calibrated one: on ECUSTFD a single item above one cubic metre carries the upper bound out to $+20$, and a percentile interval on a mean with a tail like that is known to be approximate. We do not correct it, and we do not need to: every verdict this paper reports also holds on the symmetric log error, whose distribution has no such tail (Sec. E.8), and the ECUSTFD verdict additionally holds with that item removed (Sec. 5.2). We query every VLM member using greedy decoding on the same frozen crops and a fixed JSON-only prompt (App. G.3); the prompt forms are listed in App. G.4.

Table 8: **Channel inventory.** Which fusion axes each dataset supports. A *run* is one scored prediction file on that dataset – a chain configuration (backbone, operator, viewpoint, sensor), a panel member, or a reimplemented pipeline; backbones counts the metric monocular ones. *Depth sensor*: ✓ two, • one, "–" predicted only. The VLM returns a volume without a depth field, so it never enters a backbone pool.

Dataset	Items	Runs	Backb.	Oper.	Views	Depth sensor	Prior	VLM	Cls.
ECUSTFD	837	29	4	2	✓	–	✓	✓	19
MADIMA23	82	28	4	2	–	✓	✓	✓	34
MRGBD	191	28	4	2	–	•	✓	✓	5
MetaFood3D	34	26	4	2	–	–	✓	✓	34
YCB	800	20	4	2	–	•	✓	✓	40
NOCS	13,371	26	4	2	–	•	✓	✓	5
All datasets	**15,315**	**157**							

Two channels do not always cover the same items, and when they do not, we say so. The VLM channel is scored on every released crop, with one exception we state rather than leave to the file sizes: NOCS ships 13,371 admissible rows and the panel is run on the first 1,800 of its manifest, since fourteen members at that scale would have cost more than the result could carry. Coverage also varies by member, because a model can return text that carries no number: over the panel it runs from 0.25 on YCB for the weakest member to 1.00, and the incumbent parses 87% of Nutrition5k dishes and 82% of YCB crops. Every table reports the n each cell is computed on, and the released panel file carries the attempted count beside it, so a low rate is visible rather than absorbed. We score the geometric chain only when the pipeline produced a mask and a metric depth field for that crop. On ECUSTFD, this difference is largest, and it has two causes that we keep separate. The panel scores 2,867 crops over 143 distinct targets; the geometric reference configuration produces 837 items over 70. The panel's crops also cover both viewpoints, whereas the reference configuration is a single-view (side) run, so the 2,867 crops carry roughly two rows for each geometric item. Joining the two channels row-wise would therefore weight most items twice and arbitrarily pick one of the two VLM answers; instead, we join on the chain's own viewpoint, one row per item, which is 803 of the 837. Every pool containing the VLM channel is built on that set, and every cross-channel *pool* comparison is made on it. Two kinds of tables are deliberately not: the panel of Sec. 5.3 scores each member on every crop it answered, and the per-member rows of Table 67 score it at the anchor's viewpoint, which on ECUSTFD is 1,379 items rather than 803. Each table states the count its cells are computed on, and a member can therefore carry two errors across two tables for the same dataset.

This sweep is the unit of everything that follows. Across the backbones, integration operators, viewpoints, sensors and support sources that each dataset affords under the uniform scale rule, there are 157 canonical run files over the six volume datasets, from 20 on YCB to 29 on ECUSTFD (YCB's incumbent VLM run is stored twice, under the panel's name and under the name the pool loader reads, so the files hold 156 distinct runs) (Table 8): 37 geometric layouts, 92 VLM runs and 27 runs of four published pipelines, reimplemented here and applied across datasets, as reported in Sec. 5.2 and Table 6. The VLM panel adds 21 runs on Nutrition5k and 15 on OmniFood8K, spanning its fourteen scored members (Sec. 5.3), eight datasets (YCB for the nine members that fit a

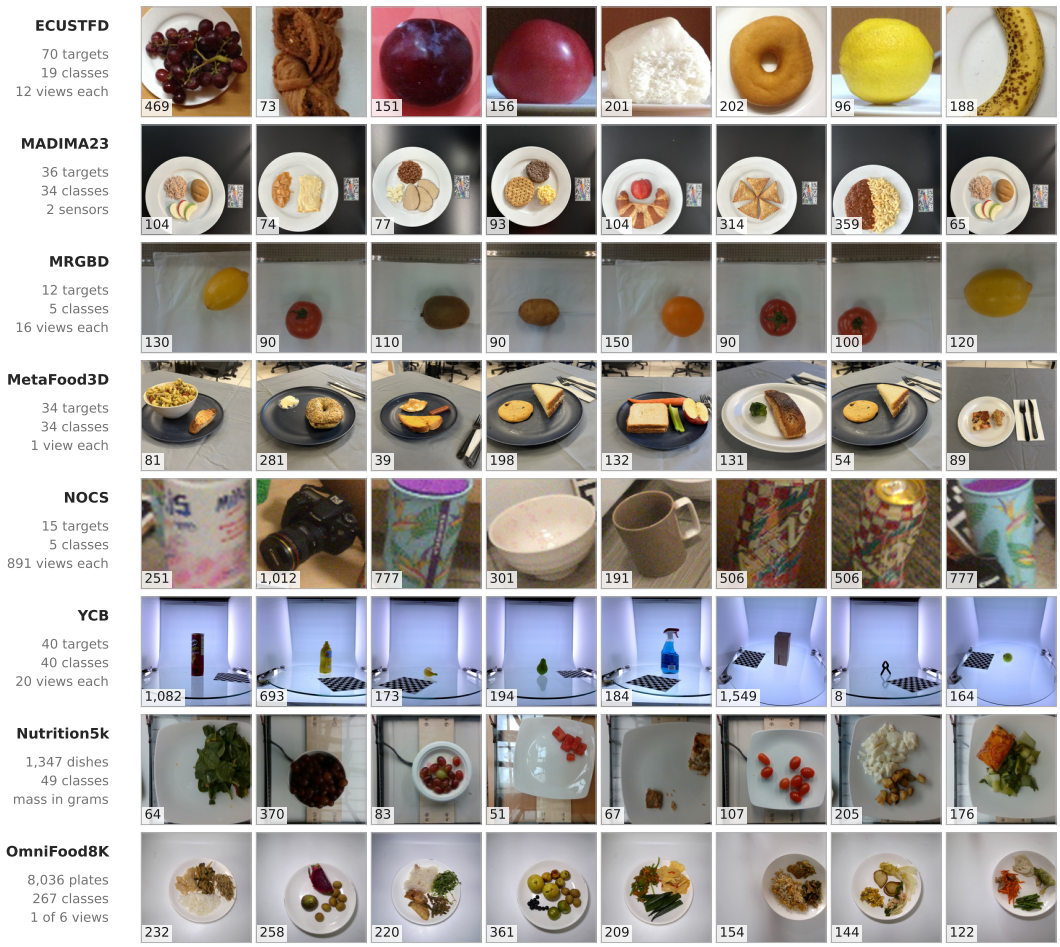

Figure 3: **What the benchmark actually sees.** Eight items per dataset, drawn from the frozen crops every source is scored on; each tile is labelled with its target value (cm^3; grams for the two mass datasets). Tiles are drawn from the full released crop set of each dataset; on ECUSTFD, this is the 143-target set the panel scores rather than the 70-target test split the chain is scored on, so a tile may lie outside the range reported in Table 1. YCB shows the rebuilt split (Sec. 5.1), and NOCS excludes the laptop class (Sec. 4).

24 GB card, added after its rebuild), four prompt forms, and three quantisation settings (bf16, 4-bit NF4, and the 235B member's released FP8). Together, these are 193 run files containing 361,275 scored per-item predictions, counting the rows the channel loader retains: those with a finite positive prediction and ground truth on a target the ground-truth audit admits. Each carries the item identifier, the prediction, the ground-truth value, the configuration that produced it, and, for the geometric runs, the per-image scale report naming the cue that fixed the scale. The released archive also retains the 438 legacy-protocol geometric runs these replace; the loader identifies and excludes them, and no number in this paper is computed from one. Three further runs sit outside this set on purpose: the component check of App. D.7 needed MADIMA23's three non-anchor backbones on its RealSense arm, which the run matrix does not open; they are kept in a separate directory the pool loader cannot reach, so that they contribute a depth measurement and not a pool. The 76 pools of Sec. 5.4, the 46 of Table 49 and the 30 control pools are drawn from the canonical runs: a pool is a set of configurations that differ along exactly one axis, or a deliberate mixture across channels.

D.5 Ground-truth audits

Two audits changed the benchmark. On NOCS, the volume is the divergence-theorem integral of the shipped mesh, which is a volume only when the mesh is closed, and none of the eighteen meshes is watertight. On the three laptop models, it fails a bound no annotation should survive: a rigid object cannot occupy 0.6 to 5% of its own convex hull, yet a laptop is recorded at 30.9 cm^3. The laptop class is excluded outright. The rule is a bound of 0.10 of the hull, set in a gap no retained class approaches: laptops fill 0.006–0.051 of their hulls, and the next class is 0.22. The remaining classes fill 0.22–0.91 (bowls 0.22–0.26, mugs 0.29–0.31, cameras 0.59–0.82, cans and bottles 0.80–0.91). The integral is taken over the mesh as shipped, in its own object frame, so for the open bowls and mugs it is the volume the shell encloses, while the chain's operators estimate the outer envelope. That gap is bounded by restricting NOCS to the three classes whose ground truth fills most of its hull: cans (0.87), bottles (0.85) and cameras (0.71). On those 8,162 items, the prior is still more accurate (0.765 against the chain's 4.498), by +3.732; however, with fifteen targets reduced to nine, the interval $[-0.34, +10.18]$ no longer excludes zero. The audit script, its threshold and the mesh repairs that failed are released.

On YCB, the released RGB and depth come from different cameras at different resolutions, and registered depth existed for one object of forty, so the usable split held 600 views of a single object and failed VARIATION. Registering the infrared depth into the RGB frame for every capture rebuilds the split to 800 items over all 40 objects: twenty captures per object, drawn without replacement at a fixed seed, spanning the rig's five camera heights and the full turntable rotation. YCB passed TARGETS only after this rebuild, which is why it stands outside the primary family.

D.6 The scale protocol, and the pipeline defects the audit caught

Every geometric number in this paper depends on the metric scale λ, so we present the protocol that fixes it first. This section is also the only place where we record the three defects in our pipeline that the audit caught, all repaired before we computed any numbers in this paper: an uninitialised depth network, fitted scale constants, and a mis-registered sensor stream on YCB.

First, the released ZoeDepth checkpoints store their weights one level deeper than the loader expects, and loading with `strict=False` matched nothing: the network silently retained its random initialisation and returned an almost constant field, 0.69 m everywhere, for every image in every dataset. Nothing downstream objected, because the second failure compensated for the first: each configuration carried a hand-set depth multiplier, fitted per backbone and per integration operator, and on two datasets per support source, and those constants had been fitted *against the degenerate fields*. A cache audit now ships with the code (zero-variance depth is refused, as is any checkpoint that leaves most parameters unloaded); the constants are gone, and the scale is recovered from the scene by the single rule of Sec. D.4. The third defect was independent of scale: YCB's infrared depth was at first unprojected through the calibration of the 1280×960 infrared image the file ships rather than through the intrinsics of the 640×480 stream it came from, which reported sensor returns on the object in 6% of captures and a sensor arm unable to produce volumes. Registering through the stream's own intrinsics (Sec. D.7) rebuilt the split from one object to forty (Sec. 5.1).

Table 9 checks that the rule lands. Four backbones whose native scales differ by up to a factor of three are given one rule; on every dataset with a physical anchor, they recover the same scene depth

to within 1.09–$1.31\times$, and that depth agrees with the dataset's own sensor or rig. Two details deserve note. The ECUSTFD coin is not clamped: its apparent diameter under each backbone *is* the measurement of that backbone's scale error (12.3 cm under ZoeDepth and 4.9 cm under UniDepthV2, medians over the side captures, against a true 2.5 cm), and a clamp of $[2.0, 3.0]$ cm would saturate on every image, turning the cue into a constant 0.83 multiplier identical across backbones, which is precisely the failure mode the protocol exists to remove. And MetaFood3D affords no anchor–no reference object, no sensor, no documented distance–so its backbones settle where their priors leave them, a $3.11\times$ spread that no rule without an anchor can shrink. Its results are read accordingly. NOCS's recovered depths sit about 20% below its own sensor because the weighted median blends the sensor cue with the tabletop priors; the blend is the rule working as specified, not an anchor failure. Nor is it what limits NOCS: the sensor arm, which uses the shipped depth itself and so carries no scale error at all, scores a mean ARE of 6.75 (median 2.07, bias $\times 3.1$) against the ZoeDepth chain's 3.29 (median 0.68, $\times 1.3$), because the sensor map is sparse and noisy on the small objects that dominate the set; removing the rule's 20% bias would move the chain by a factor of 1.25^3 at most and would not change its verdict against the prior (0.52).

The constants did not purely add noise; they determined the results. Under per-view constants, the ECUSTFD side capture appeared stronger, and under the uniform rule, the ranking reversed (Sec. 5.4). Under per-support constants, the segmentation axis appeared inert because each segmenter's constant absorbed its own mask bias, whereas under the uniform rule that axis carried the two largest fusion gains in the study. With degenerate depth fields, the chain matches a human crowd on MetaFood3D, a parity a working pipeline does not reproduce (Sec. F.17). The same defect shows in the uncertainty: a depth network that returns one constant field for every view makes a target's predictions perfectly correlated, and on NOCS it inflates the target-clustered interval over the row-level one by $26\times$, near the ceiling of about 30 that the square root of 891 views per target sets; the working network varies across views and the inflation falls to $8.1\times$. A benchmark that permits per-cell calibration does not measure its axes; it measures its calibration.

Table 9: **One scale rule, four backbones, one scene depth.** Median in-mask scene depth (m) recovered by each backbone under the uniform protocol, at the reference chain of every dataset. The reference is measured from the dataset itself – its shipped sensor depth or its documented rig distance – never from volume ground truth. *Spread* is the ratio of the largest recovered depth to the smallest. Sec. D.6 reads the check.

Dataset	**Anchor**	**Ref. (m)**	ZoeDepth	Depth Pro	MoGe-2	UniDepthV2	**Spread**
ECUSTFD	coin, 2.5 cm	–	0.35	0.32	0.34	0.34	1.09×
MADIMA23	rig, documented	0.50	0.43	0.49	0.48	0.52	1.22×
MRGBD	own sensor	0.25	0.22	0.21	0.20	0.26	1.31×
MetaFood3D	none (prior only)	–	0.33	0.94	1.04	0.79	3.11×
YCB	own sensor	0.95	0.86	0.84	0.79	0.79	1.09×
NOCS	own sensor	1.02	0.87	0.80	0.80	0.77	1.12×

D.7 The two upstream stages, against the truth their datasets ship

Every other measurement in this paper reaches the depth map, and the mask only through the volume they produce – a blunt instrument: a scale error, a shape error, and a mask error all arrive as one number. Four datasets let us do better. MRGBD, NOCS, YCB and MADIMA23 ship a sensor depth map and a per-object mask alongside every image, so the pipeline's own cached predictions can be scored against them directly. The predicted depth is read from the cache the runs consumed and multiplied by the per-image scale those runs applied, and the predicted mask is addressed by the exact cache signature its run wrote: nothing is re-estimated and nothing is re-fitted. Depth is scored inside the annotated mask, so segmentation cannot contribute to the depth number. ECUSTFD ships bounding boxes only, and MetaFood3D ships neither depth nor masks, so the two datasets carrying much of Sec. 5.2 cannot be checked this way.

Depth. Table 10 and Fig. 4 report it. The depth maps are structurally good and metrically mediocre, and nearly all of the difference is captured by a single number per image. The in-mask absolute relative depth error (AbsRel) spans 0.041 to 0.376 across the sixteen scored configurations; applying the single multiplier that would have been optimal for that image reduces it to between 0.008 and 0.037 everywhere (Fig. 5), a median of 11% of the original error on MADIMA23 and on NOCS, 12% on YCB and 44% on MRGBD. What the backbones get wrong is not the shape of the scene, which they recover to within a few percentage points, but its size, and the residual error lives in the

scale protocol of Sec. D.6. Two configurations show this most starkly. On NOCS, the recovered depth sits 15–27% below the sensor under all four backbones, enough to take δ_1, the share of pixels within 25% of the sensor depth, from 0.885 under ZoeDepth to 0.012 under MoGe-2, while the error left after rescaling stays at 0.021–0.037 for both. UniDepthV2 on MADIMA23 is even more extreme: an AbsRel of 0.376 and a δ_1 of exactly zero, figures that would ordinarily indicate a broken estimator, sit on top of one of the two lowest residuals in the table (0.010; only YCB's 0.008 is smaller). The field is correct to one percentage point but is placed at the wrong distance.

Two further readings bear on the rest of the paper. First, YCB's registered depth is sparse, and the sparsity stems from the rig, not the data. Its infrared stream is 640×480 against a 1280×1024 RGB frame, so projecting one into the other reaches at most one target pixel in four; the registered map covers a median 15% of the frame, two thirds of that ceiling, and clears the 200-pixel floor inside the object mask in 752 of 800 captures. That floor is a reporting criterion for this audit rather than a pipeline gate: a run scores every capture, using whatever registered depth falls inside the mask, so the 48 sparse captures are in the YCB numbers everywhere else in the paper. Removing them does not move the verdict, which is the point of naming them: on the 752 that clear the floor, the chain reads 10.951 against the global prior's 1.439, a paired difference of $+9.512$ $[+3.165, +17.612]$ against $+15.370$ $[+4.071, +31.550]$ over all 800. The projector's baseline shadow accounts for the rest, leaving the raw map a median 0% valid in its leftmost tenth (Fig. 4). Registration must use the intrinsics of the 640×480 stream itself, not those of the 1280×960 infrared image the file also ships (Sec. D.6). The depth and mask checks of Table 10 read 900 frames per dataset and backbone, taken at a uniform stride through the image list rather than at random, which is enough for a median to settle without re-reading the whole corpus. Second, depth accuracy does not order volume accuracy. Across the sixteen configurations, the two are essentially unrelated (Spearman $\rho = 0.14$, $p = 0.59$), and ranking configurations within their own dataset and pooling the ranks turns the association negative ($\rho = -0.25$, $p = 0.35$): three of the four datasets relate negatively (-0.8 on NOCS and on YCB, -0.4 on MADIMA23) against one perfect positive ($+1.0$ on MRGBD). Choosing a backbone by its AbsRel would not have chosen the backbone with the lowest volume error.

Masks. Table 10 and Fig. 6 present them and correct how Sec. 5.4 should be read. The default segmenter is accurate wherever the annotation is a single object: median IoU 0.979 on MRGBD, 0.925 on MADIMA23, 0.886 on NOCS and 0.842 on YCB, with accuracy above 0.96 throughout. Boundary agreement is markedly weaker: measured on the silhouette contour within two pixels, F1 falls to 0.965, 0.751, 0.822 and 0.508 on the same four datasets. On YCB, a segmenter that gets 84% of the silhouette right gets barely half of the rim right, and it is the rim rather than the interior that determines the volume of a thin or shallow object.

The second support source is not a second segmenter of the same kind. BiRefNet is a single-salient-object model, and on the multi-object NOCS scenes it returns one mask per image – a median of 94,776 pixels, some 30% of the frame, spanning the whole cluster of objects together with the fiducial markers around them – against a median annotated object of 4,300 pixels. The pipeline hands one mask to each of the 4.86 retained objects in the average frame (13,371 objects over 2,754 frames; 5.85 with the excluded laptops), so BiRefNet reaches 17% of them, and its median per-object IoU is 0.000. On YCB, with one object per image, it is a genuine, if weak, segmenter: median IoU 0.374, recall 0.861 against precision 0.398, and 36 captures (4.5%) on which it returns no mask, so the run records zero volume. So the support pools of Sec. 5.4 are not two comparable segmenters whose failures happen to differ. They are an object segmenter set against a scene-level saliency mask, and what the learned gate realises there is largely the gate learning when the object fills enough of the frame for the saliency mask to be usable. The fused numbers stand as reported; the mechanism is narrower than complementarity between peers, and the distinction belongs in the record.

Mask quality then explains remarkably little of the downstream volume error. Over the 5,445 objects for which an annotated mask and an anchor-configuration volume both exist, per-object IoU and volume ARE relate at $\rho = -0.14$: the median error is 0.811 below IoU 0.5 and 0.418 above IoU 0.95, a factor of two against volume errors spanning three orders of magnitude. On three of the four datasets, in short, both upstream stages are accurate enough that neither is the binding constraint; the metric scale binds, and no single image determines that scale. YCB is the exception, and the one that matters most, since it carries a corrected verdict. Its recovered scale is within 6% of the sensor, and its mask IoU is 0.84, yet the chain's median ratio is $\times 5.4$, and the per-class median runs from $\times 1.2$ (mustard bottle) to $\times 60$ (adjustable wrench). The error is a small-object artefact of the single-view

hull. The depth range inside the mask, 0.133 m at the median, is more than twice the median object extent (0.057 m), because the mask pixels that miss the object fall on the table behind it and the hull then spans table and object; log size, in-mask depth range and mask area together explain 88% of the variance of the log error, and the smallest quartile of objects (median 57 cm^3) carries a median ratio of $\times 4.2$ against $\times 0.85$–0.98 for the other three. TSDF ($\times 3.0$) does not repair it, and the sensor arm, whose sparse map misses most of a small object, underestimates instead ($\times 0.19$). The YCB verdict of Sec. 5.2 therefore reads as the global prior beating a chain whose operator cannot handle small objects in one view, a defect of the chain rather than of its scale.

Table 10: **The two upstream stages against the ground truth their datasets ship.** *Left, depth:* median per-image error inside the annotated mask, so segmentation cannot contribute; $s^\star$ is the per-image multiplier that would have been optimal, and AbsRel_s the error remaining once applied. *Right, masks:* median over objects; *Found* is the fraction of annotated objects given any mask (a miss scores zero and is counted), and boundary F1 is measured within two pixels of the silhouette contour. Sec. D.7 reads both halves.

Dataset	Backbone	n	AbsRel↓	δ_1↑	$s^\star$	AbsRel_s
MRGBD	ZoeDepth	192	0.072	1.000	0.94	0.031
	Depth Pro	192	0.041	1.000	1.00	0.023
	MoGe-2	192	0.062	1.000	1.06	0.028
	UniDepthV2	192	0.231	0.466	0.80	0.027
NOCS	ZoeDepth	900	0.152	0.885	1.18	0.037
	Depth Pro	900	0.241	0.093	1.31	0.024
	MoGe-2	900	0.242	0.012	1.31	0.021
	UniDepthV2	900	0.260	0.019	1.36	0.029
YCB	ZoeDepth	752	0.078	1.000	0.94	0.013
	Depth Pro	752	0.081	1.000	0.98	0.010
	MoGe-2	752	0.076	1.000	1.02	0.008
	UniDepthV2	752	0.082	1.000	1.02	0.010
MADIMA23	ZoeDepth	40	0.195	0.642	1.12	0.022
	Depth Pro	40	0.129	1.000	1.09	0.014
	MoGe-2	40	0.124	1.000	1.09	0.013
	UniDepthV2	40	0.376	0.000	0.90	0.010

Dataset	Segmenter	n	Found↑	IoU↑	Prec.↑	Rec.↑	BF1↑
MRGBD	GDINO+SAM	192	1.00	0.979	0.999	0.981	0.965
NOCS	GDINO+SAM	5,267	0.83	0.886	0.965	0.915	0.822
	BiRefNet	5,267	0.17	0.000	0.000	0.000	0.000
YCB	GDINO+SAM	800	0.91	0.842	0.976	0.867	0.508
	BiRefNet	800	0.75	0.374	0.398	0.861	0.304
MADIMA23	GDINO+SAM	87	0.90	0.925	0.989	0.938	0.751

D.8 Depth and mask galleries behind the component check

The three figures below accompany Sec. D.7: the predicted depth beside the depth each dataset ships (Fig. 4), how much of the depth error a single per-image rescale removes (Fig. 5), and the predicted masks against the annotated ones (Fig. 6).

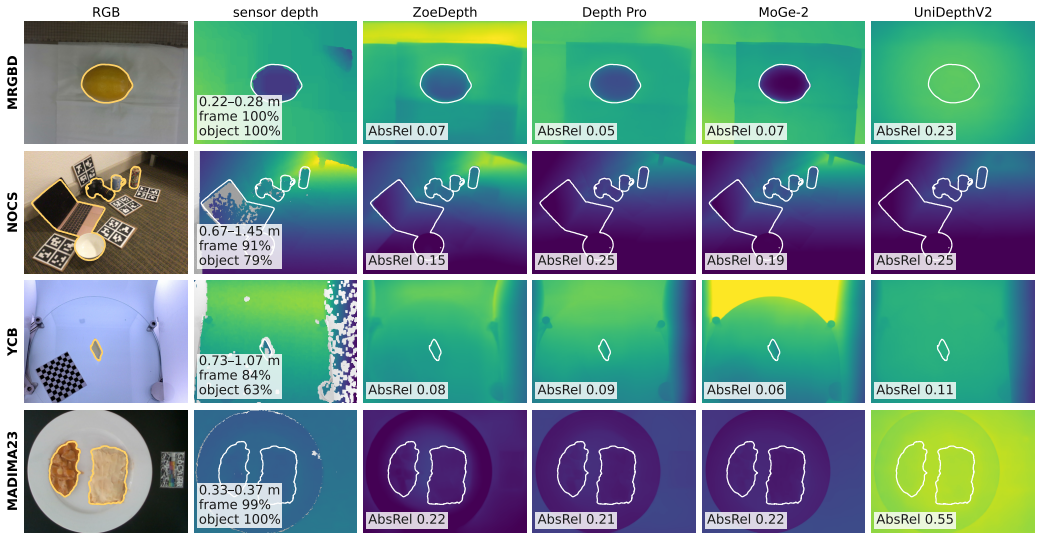

Figure 4: **Predicted depth alongside the depth the dataset ships.** One frame per dataset, the frame at that dataset's median in-mask error, so every row is a typical case. All panels in a row share the sensor map's colour scale: a wrong scale reads as a wholesale colour shift, a wrong structure as a changed pattern, and almost everything visible here is the former. Sensor panels paint returns over a washed-out copy of the frame; YCB's sparse row is drawn and framed as its rig requires (Sec. D.7).

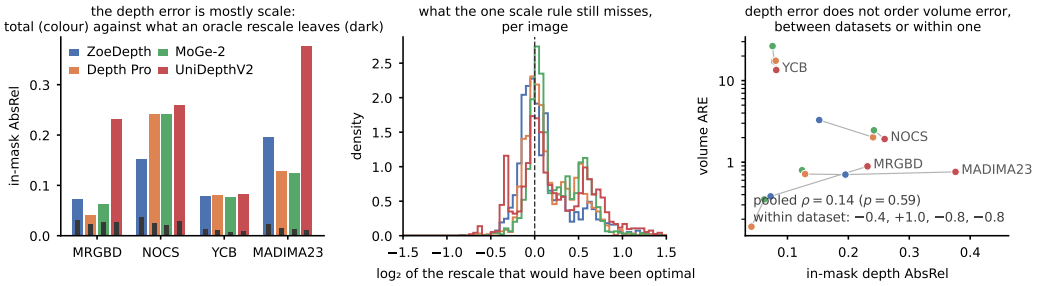

Figure 5: **How much of the depth error a single number would remove.** Left: median in-mask AbsRel per configuration, the error surviving the optimal per-image rescale in dark grey. Middle: the distribution of that rescale. Right: depth error against volume error, joined within datasets – unrelated pooled, negatively related within three of the four.

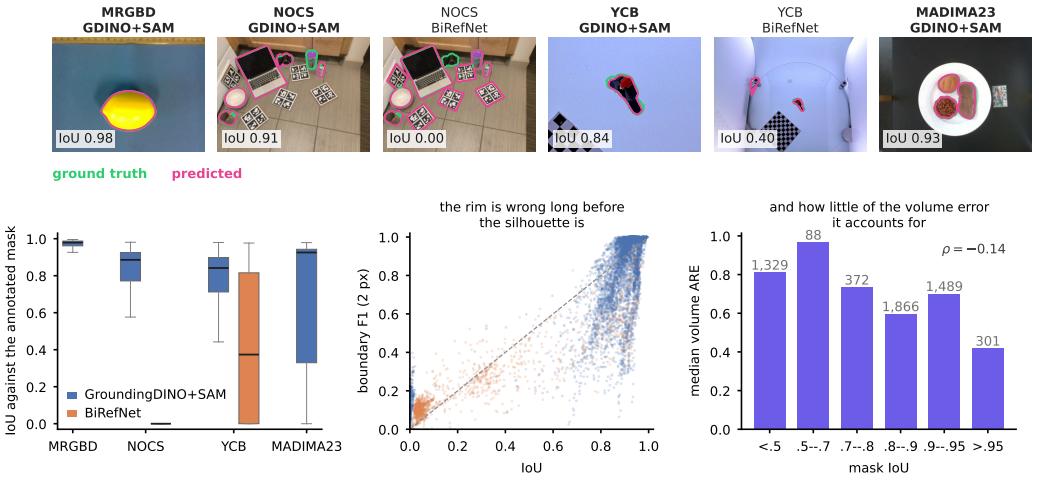

Figure 6: **Predicted masks against annotated masks.** Top: each dataset's median-IoU frame, annotation in green, prediction in pink; on NOCS, BiRefNet returns one salient region spanning the whole scene. Bottom: per-object IoU; boundary F1 against IoU (the rim is wrong long before the silhouette); and median volume error by IoU band.

D.9 Reproducibility and explainability

Every table and figure in this paper is regenerated from the released per-item prediction files using the scripts in the repository, and every number is traceable to a row in those files: for each item, the artefacts record each source's prediction, its ground truth, its category, its instance key, and the pre-integration cues used by the learned selector. Fold assignments, bootstrap draws, and the shuffled-cue control use fixed seeds. Each criterion is evaluated against the stated threshold rather than judged, so you can assess a new dataset before a single geometric run. Where a fusion rule wins, the released pool diagnostics (the pool's pairwise error-correlation matrix, its member list, and its best single member, alongside the per-item predictions from which the oracle's choice is recoverable) show *which* source it drew on and *why*: the explanation is the pool's disagreement structure, not an opaque score.

The NOCS data used here are the REAL275 real-capture test set; we retain the name NOCS in the tables for continuity with our released artefacts, but we evaluate no synthetically rendered NOCS images in this paper.

D.10 Using and extending OneVol

What it is for. OneVol is an evaluation benchmark over published datasets. Its task is to estimate, for each evaluated item, one positive scalar in the dataset's unit (cm^3, or grams on the two mass datasets), and its question is what a source adds to the class prior. It is intended for comparing sources and fusion rules under one protocol, not for certifying an estimator for dietary or clinical use: five of the six volume datasets fail at least one validity criterion, and Table 1 states which claim each one licenses.

Scoring a new method. A contributor writes one CSV per dataset with a row per evaluated item and four columns: `id`, the item's identifier as the dataset distributes it; `class`, the category the class prior is keyed on; `V_pred_cm3`, the estimate; and `V_gt_cm3`, the ground truth in the same unit (the column names are historical on the mass datasets). The command `python -m fusion.score --pred file.csv --dataset name`, which the supplementary snapshot exercises on a worked submission (`examples/submission_ecustfd.csv`, the reference chain in the same format, returning this paper's ECUSTFD numbers), then reports, on folds grouped by target, the dataset's five validity criteria, the method against the class and global priors on mean ARE and on the symmetric log error with paired target-clustered 95% intervals, and what the method adds to the prior under the class-anchored rule, with the fusion condition of Eq. (4) evaluated first and the number of targets the gain would need to resolve (App. F.11) stated beside the number the file holds, so that a benchmark too small to carry the measurement says so on the first run. A target is identified by an optional `target_id` column and otherwise by its distinct ground-truth value (Sec. 4). The command refuses a file that lacks a required column and reports how many rows it read and how many it scored, since rows with a non-finite estimate, a non-positive ground truth, a repeated identifier or an excluded class are dropped. The released per-item files of every source in this paper are in the same format, so any of them reproduces its row of Table 63 with that one command.

Adding a dataset. The same command accepts any dataset name and evaluates the criteria on the file it is given, so a new dataset can be assessed from its ground truth and class labels alone, before any geometric run. Adding it to the paper's tables takes its name and unit in the loader and one reference configuration.

What the snapshot holds. The supplementary material contains the benchmark code, the per-item predictions of every geometric source and of the fourteen-member panel, the analysis tables behind every number, and the registered plans as frozen. From a clean copy, with only NumPy, pandas, SciPy, scikit-learn and Matplotlib installed, the analysis scripts recompute their result files from the per-item predictions on a CPU (`fusion.strongest_opponent` in seconds, the main sweep `fusion.run_all` in more than ten minutes; rerun from the snapshot, it reproduces the chain, prior and paired-difference values of Table 18 exactly), and `python -m fusion.tables` then formats every table of the paper from those result files; it does not itself recompute them. The ground-truth defects of Sec. D.5 are applied at load time and listed in the snapshot's README, so a user cannot score the excluded NOCS class by accident.

D.11 BENCHMARK CARD

Motivation. To separate what recognition and what geometry contribute to a monocular volume estimate, and to test decision-level fusion where a strong prior sits beside a weak measurement, with ground truth in physical units (Sec. 1).

Composition. No new images. The release holds the benchmark code; per-item predictions of 37 geometric configurations on six volume datasets and of a fourteen-member vision–language panel on eight; the analysis tables behind every number; and the registered plans as frozen. Rows, targets, classes and failed criteria per dataset are in Table 1; every run is inventoried in App. D.4.

Collection. None, except the human reference: seventeen volunteers, nineteen published MetaFood3D scenes, numeric estimates only, no identifiers (Ethics statement; App. F.17).

Processing. Ground-truth volumes are the publishers', except YCB and NOCS, whose meshes we integrate. One rule sets the metric scale for every chain, with no constant fitted to the evaluation target (App. D.6). A target is identified by its distinct ground-truth value (Sec. 4).

Known defects. No NOCS mesh is watertight, and its laptop class is excluded by a stated convex-hull rule. YCB shipped registered depth for one object of forty, and its split is rebuilt (App. D.5). On 16 ECUSTFD items the coin cue saturates its clip bound, and the reported numbers are the pipeline as released (App. I.5). MetaFood3D supplies no physical scale cue. Five of the six volume datasets fail at least one validity criterion (Sec. 5.1).

Intended use. Comparing sources and fusion rules under one protocol, and assessing whether a dataset can support such a comparison. It is not a certification of any estimator for dietary or clinical use (App. D.10).

Distribution and licence. During review, the anonymised supplementary snapshot. Our code, predictions and analyses are under CC BY 4.0. The datasets' images and meshes and the evaluated models' weights are not redistributed; the ground-truth values and item identifiers that the per-item files repeat remain under their publishers' terms, several of which are non-commercial or unstated (Table 11).

Maintenance. After review the artefacts are distributed as tagged releases of the project's public repository, each with a manifest that lists every file with its checksum; a correction is issued as a new tagged release and earlier ones stay available.

Table 11: **Provenance and terms of the source datasets.** Terms are quoted or summarised from each publisher's own page as read in September 2026; "none stated" means the source carries no licence text. OneVol redistributes none of these datasets' images or meshes. Its per-item files repeat their ground-truth values and item identifiers so that predictions can be scored, and those values remain under the terms listed here.

Dataset	Source	Terms as stated by the publisher
ECUSTFD	`github.com/Liang-yc/ECUSTFD-resized-`	none stated ("a free public food image dataset"; citation requested)
MADIMA23	`huggingface.co/datasets/lubnaa25/Madima23` (access on request)	Apache License 2.0
MRGBD	distributed with Sari & Gofuku (2023)	none stated
MetaFood3D	`lorenz.ecn.purdue.edu/~food3d`; the 34-item challenge release used here: Kaggle, "3D Reconstruction From Monocular Multi-Food Images"	CC BY-NC 4.0, non-commercial research only; the challenge release states no data licence beyond the competition rules
YCB	`ycbbenchmarks.com`	data CC BY 4.0, code MIT
NOCS REAL275	`github.com/hughw19/NOCS_CVPR2019`	non-commercial use only, citation required (code MIT)
Nutrition5k	`github.com/google-research-datasets/Nutrition5k`	CC BY 4.0
OmniFood8K	`yudongjian.github.io/OmniFood8K-food`	none stated

D.12 COMPUTE AND CARBON IMPACT

The reported runs took 94 hours on a single NVIDIA RTX 3090 (350 W), including the label-parity rerun, and 107 GPU-hours, read from the job logs, on NVIDIA H100 nodes (700 W) of a national supercomputer, which hosted the larger panel members; scoring the fourteen-member panel on OmniFood8K accounts for 60 of the latter. Charging accelerators at their rated draw, with a power usage effectiveness of 1.5 for the workstation and 1.1 for the supercomputer, yields 131 kWh, or 34 kg CO_2e under the 2025 electricity mix published by the national energy regulator(0.258 kg CO_2e per kWh). Exploratory runs are not counted. Because we release per-item predictions, every table and figure here regenerates on a CPU in about an hour.

E REPLICATION, GROUPING AND PROTOCOL VARIANCE

This appendix gives the validity verdicts and design quantities for each benchmark, and then measures what the protocol itself contributes: the grouping of folds, their number, and their assignment.

E.1 DATASET VALIDITY, CRITERION BY CRITERION

Five of the six volume datasets fail at least one criterion (Table 1). Three fail a data-related criterion. MRGBD and NOCS fail TARGETS, with 12 and 15 distinct objects behind 191 and 13,371 rows; MRGBD's twelve objects resample as nine groups, because three of its volumes recur across class pairs, which widens rather than narrows its intervals. ECUSTFD fails SPREAD at 0.098 against the nominal 0.10 on its test split, close enough to the threshold that App. C.7 sweeps it; the 143 targets the panel scores would pass at 0.134. MADIMA23 and YCB fail only SKILL, which a better chain would flip. MetaFood3D passes every criterion that applies to it, but SPREAD is inapplicable there because each class holds one target, and it passes SKILL on a mean-ARE difference whose interval spans zero and which reverses under the log error. No volume dataset therefore clears SKILL on both metrics. Nutrition5k and OmniFood8K meet all four data criteria. SKILL, evaluated on the *incumbent* VLM, Qwen2.5-VL-7B, fixed before the panel was assembled, fails on both.

Clustering the bootstrap on the target instead of the row widens NOCS's interval on the reference chain from $[2.90, 3.70]$ to $[0.87, 7.34]$, an inflation of $8.1\times$; MRGBD widens by $2.3\times$ and YCB by $1.5\times$. The ceiling on this inflation is the square root of the rows per target, about 30 on NOCS. It is approached when every view of a target fails alike: an uninitialised depth network that returned one constant field per view produced $26\times$ (App. D.6). Any protocol that resamples rows understates its uncertainty by a factor that grows with view multiplicity, and most do so when the pipeline is most broken.

E.2 WHAT EACH BENCHMARK SUPPLIES

Table 12: **What each benchmark supplies before any method is run.** σ_w is the pooled within-class spread of $\log V$ over distinct targets, the error a class prior cannot go below (Sec. 3.2); ***Prior (OOF)*** is the class prior's out-of-fold log RMSE, and the ratio shows how far σ_w understates it. ***DEFF*** is Kish's design effect of the chain-minus-prior log difference with unequal clusters; ***Effective rows*** is rows over DEFF with a 95% target bootstrap interval; the last column is the number of distinct targets needed to detect a 0.10 log-error difference at 80% power. §, fails TARGETS.

Dataset	Rows	Targets	σ_w	Prior (OOF)	Ratio	DEFF	Effective rows	Targets for 0.10
ECUSTFD	837	70	0.210	0.228	1.09	2.9	286 [146, 575]	69
MADIMA23	82	36	–	–	–	2.0	41 [36, 52]	785
MRGBD§	191	9	0.093	0.143	1.54	34.5	6 [3, 20]	41
MetaFood3D	34	34	–	–	–	–	34	712
YCB	800	40	–	–	–	10.8	74 [63, 98]	1,008
NOCS§	13,371	15	0.397	0.552	1.39	289.8	46 [26, 141]	139
Nutrition5k	1,347	1,347	0.558	0.576	1.03	–	–	–
OmniFood8K	8,036	8,036	0.324	0.343	1.06	–	–	–

Table 12 evaluates the measurement model's design quantities per dataset. The class prior's out-of-fold log RMSE exceeds σ_w by 3–9% where classes hold many targets (ECUSTFD, Nutrition5k, OmniFood8K). On NOCS and MRGBD, whose classes hold two or three targets, it exceeds it by

39–54%, since a fold removes a large share of each class. σ_w is undefined where fewer than half the classes hold two targets (MADIMA23, MetaFood3D, YCB). Used as a classifier (a model beats the prior when its own log error is below σ_w), the bar is right in 63 of 66 model–dataset cases. That is not detectably better than always backing the prior (58; exact McNemar $p = 0.06$), and the call is nearly circular, because the channel's required error is almost its out-of-fold outcome (Table 14). We therefore use σ_w as a bar, not as a classifier.

The replication columns price the rows. ECUSTFD's twelve views per target are worth about four independent rows each, and NOCS's 891 about three, so its 13,371 rows carry the evidence of 46 (95% interval 26–141). Detecting a 0.10 difference in log error at 80% power needs 41 targets on MRGBD, 69 on ECUSTFD, 139 on NOCS and 712–1,008 on MetaFood3D, MADIMA23 and YCB, a twenty-five-fold spread. These figures belong to one contrast, the reference chain against the prior, and depend on that chain; and on NOCS any interval still rests on 15 clusters, 14 degrees of freedom, whatever the effective row count.

E.3 NOCS WITH THE OBJECTS OF ITS TRAINING SPLIT

NOCS fails TARGETS because its evaluation split holds 15 admissible objects. REAL275's training split holds 18 further real objects, three per class, so we ran the released chain on it, unchanged, with the four backbones and the flags of the evaluation runs, on 280 frames drawn at random (1,253 object crops; the design effect of Sec. 5.1 says rows beyond a handful per target add almost nothing). With the laptop class excluded as in the paper's loader, the union holds 30 targets, six per class, and passes VARIATION, TARGETS, CLASSES and SPREAD; it fails SKILL. The data, the frames, the sources, two hypotheses and their criteria were written down, with the file's hash, before the first run on these scenes. Nothing was deposited with a third party, so this is a pre-specified exploratory analysis and not a registration.

The first hypothesis holds (Table 13). On the union the class prior is more accurate than the reference chain by $+0.379$ in log error (95% CI $+0.153$ to $+0.569$) and on 26 of the 30 targets, and the interval excludes zero for every backbone. NOCS therefore moves from a direction to a comparison, and the comparison agrees with the direction it had.

The second does not meet its criterion. The class-anchored rule lowers the prior's log error by $+0.0198$ with the reference chain, an interval that excludes zero ($+0.0032$ to $+0.0383$), it wins all 30 partitions, the fusion condition passes, and the predicted log-RMSE ratio of 0.964 is realised at 0.968. But the criterion written down was the one-sided within-class permutation test at 0.05, and it gives $p = 0.0712$. No other backbone reaches it either. We report the hypothesis as not supported.

The laptops. The plan excluded the laptop class by the loader's rule and promised a check of whether the rule's reason holds for the new meshes. It does not: the paper's audit, applied unchanged, admits all three training-split laptops (hull fractions of 0.35 to 0.52 against the 0.10 threshold that the evaluation split's laptops fail at 0.006 to 0.051). Including them, a choice made after the fact, gives 33 targets, a gain of $+0.0257$ and $p = 0.0365$. We show that row for completeness and draw no conclusion from it.

Table 13: **NOCS with the objects of its training split added.** The released chain, unchanged, run on 1,253 crops of REAL275's real_train; the union holds 14,485 items over 30 targets and meets TARGETS. Log error of the chain and of the class prior on the paper's folds; their difference with a target-clustered 95% interval (**bold** excludes zero); targets on which the prior is the more accurate side; the class-anchored gain with its interval; its one-sided within-class permutation p (10,000 draws on the union rows, 1,000 otherwise); and the predicted / realised log-RMSE ratio. *, the reference chain, on which the plan's two hypotheses were stated. ‡, post hoc: the three real_train laptops, which the paper's mesh audit admits. §, 15 targets, directions only.

Sample	Backbone	Targets	Chain	Prior	Chain − prior [95% CI]	Prior wins	Anchored gain [95% CI]	Perm. p	Ratio pred. / real.
union	ZoeDepth*	30	0.803	0.424	**+0.379** [+0.153, +0.569]	26/30	**+0.0198** [+0.0032, +0.0383]	0.0712	0.964 / 0.968
union	Depth Pro	30	0.813	0.424	**+0.389** [+0.151, +0.602]	27/30	+0.0109 [-0.0100, +0.0310]	0.1548	0.970 / 0.988
union	UniDepthV2	30	0.931	0.424	**+0.507** [+0.300, +0.714]	26/30	+0.0027 [-0.0162, +0.0219]	0.2771	0.975 / 0.981
union	MoGe-2	30	0.831	0.424	**+0.407** [+0.169, +0.612]	27/30	+0.0107 [-0.0133, +0.0329]	0.1527	0.967 / 0.989
union + train laptops‡	ZoeDepth*	33	0.804	0.430	**+0.374** [+0.143, +0.573]	27/33	**+0.0257** [+0.0087, +0.0440]	0.0365	0.968 / 0.955
real_train alone§	ZoeDepth*	15	0.671	0.430	+0.241 [-0.156, +0.563]	12/15	+0.0064 [-0.0400, +0.0497]	0.0340	0.941 / 0.956
real_test alone§	ZoeDepth*	15	0.814	0.431	+0.383 [+0.146, +0.585]	13/15	+0.0149 [+0.0016, +0.0284]	0.0989	0.977 / 0.979

E.4 THE SUBSTITUTION CONDITION, MODEL BY MODEL

The design quantity σ_w of Table 12 can be turned into a per-model call: predict that a channel beats the class prior when its bias-corrected log-error spread σ_s is below σ_w. Table 14 scores that call on its own loss, the out-of-fold log RMSE of the bias-corrected channel against that of the out-of-fold prior. On the geometric channel there is almost nothing to call: of twelve defined cells, the chain beats the prior in one, MRGBD with MoGe-2 (0.137 against 0.143), and that is the one the call misses, so it scores 11 of 12, the base rate. On the semantic channel, each member scored once per dataset on its direct prompt; a member beats the prior in 8 of 66 defined cells, all on the two mass datasets. The call is right in 63 of 66 versus 58 for always backing the prior, with 5 versus 0 discordant cells (exact McNemar $p = 0.06$). The cells within a dataset share σ_w and the prior, so they are not independent, and σ_s differs from the channel's own out-of-fold error by 0.9% at the median cell ($r = 0.998$), so what a call tests is only how well σ_w stands in for the out-of-fold prior. We therefore report σ_w as a bar, not a classifier.

Table 14: **The substitution condition, scored model by model.** The channel is predicted to beat the class prior when its log-error spread σ_s (after a global bias) is below the benchmark's within-class spread σ_w, pooled over classes with each class weighted by its degrees of freedom and computed over distinct targets. σ_w is a design-stage quantity; σ_s is not, since it needs ground truth, and it nearly equals the channel's own out-of-fold error, so what a call tests is how well σ_w stands in for the out-of-fold prior, one ratio per dataset (*Prior (out of fold)* against σ_w). The outcome is the out-of-fold log RMSE of the bias-corrected channel against that of the out-of-fold class prior. Undefined where fewer than half of the classes hold two distinct targets. (a) The geometric chain, one cell per backbone; (b) the semantic channel, one cell per panel member on its direct prompt, prompt variants excluded, unparsed answers dropped row by row and the NOCS laptop class excluded as in Sec. 4. The cells within a dataset share σ_w and the prior, so they are not independent. [§], fails TARGETS.

(a) Geometric channel

Dataset	σ_w	Prior (out of fold)	σ_s range	Channel beats prior	Called
ECUSTFD	0.210	0.228	0.756–1.177	0/4	4/4
MADIMA23	–	0.650	0.933–1.304	0/4	*undefined*
MRGBD[§]	0.093	0.143	0.135–0.370	1/4	3/4
MetaFood3D	–	1.101	1.366–1.821	0/4	*undefined*
YCB	–	1.310	1.770–1.944	0/4	*undefined*
NOCS[§]	0.397	0.552	1.028–1.157	0/4	4/4
All defined cells				1/12	**11/12**
Always guessing the commoner outcome					11/12
Exact McNemar test against it					$p = 1.00$

(b) Semantic channel

Dataset	σ_w	Prior (out of fold)	σ_s range	Channel beats prior	Called
ECUSTFD	0.215	0.236	0.830–1.960	0/13	13/13
MADIMA23	–	0.658	0.612–1.390	2/11	*undefined*
MRGBD[§]	0.093	0.143	0.174–1.865	0/13	13/13
MetaFood3D	–	1.101	1.051–3.123	1/11	*undefined*
YCB	–	1.335	1.654–2.132	0/9	*undefined*
NOCS[§]	0.397	0.551	0.560–1.760	0/12	12/12
Nutrition5k	0.558	0.576	0.495–1.531	5/14	12/14
OmniFood8K	0.324	0.343	0.313–1.766	3/14	13/14
All defined cells				8/66	**63/66**
Always guessing the commoner outcome					58/66
Exact McNemar test against it					$p = 0.06$

E.5 INSTANCE-LEVEL GROUPING IS A PREREQUISITE

Two independent components of this benchmark, the category prior of Sec. 3.1 and the learned selector of Sec. 3.1, can be implemented with cross-validation folds grouped by *row*, the natural default when each row is one evaluated image. In this domain, that default is incorrect by a large factor,

because a single physical target contributes many rows (top and side captures, sensor distances, meal contexts, rendered viewpoints). Under row-level grouping, a target appears in both training and test, so a prior can memorise its volume and a selector can learn which source works for it. The distinction is recognised in the field, which separates a *food type* from a *food item*, an instance within that type (Vinod et al., 2024); what has not been measured is how much the reported numbers change when the folds ignore it.

App. E.6 quantifies both. Where a per-class prior can be learned across instances, row-level grouping overstates it by $1.8\times$ on ECUSTFD, $1.9\times$ on NOCS and $4.4\times$ on MRGBD. MADIMA23 is a different case: each target appears in only two or three rows, so under row-level grouping the prior reads the target's own volume from the other row (0.081), while under instance grouping it collapses towards the global median (0.436), a $5.4\times$ gap that reflects degeneracy rather than a learnable prior. The selector side is weaker evidence: the sign change App. E.6 reports occurs only under the fitted-constant protocol of App. D.6, and under the uniform rule the nested-selected rule on that pool, the plain mean, captures the same headroom under either grouping, so the grouping claim rests on the prior and the bootstrap. The prior columns above still separate by up to $5.4\times$, and the failure-prediction forest below still collapses from perfect to modest the moment targets are held out. Fig. 7 brings the three together on one page.

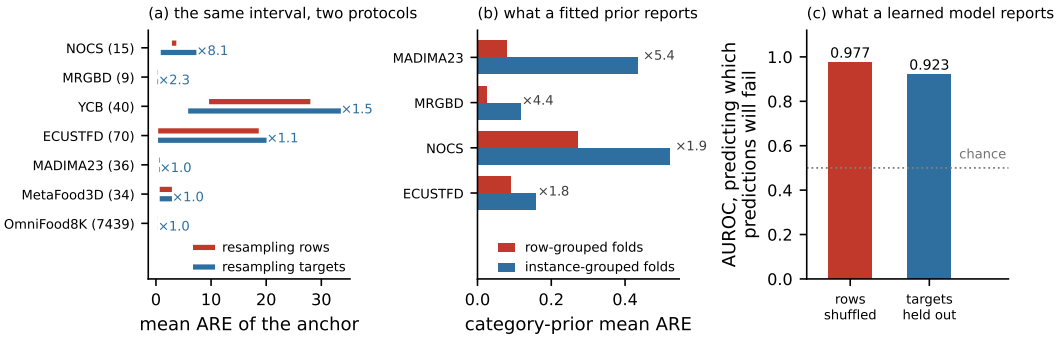

Figure 7: **One decision, three components, the same direction.** What changes when a physical target may appear on both sides of a split. (a) Row against target-instance resampling of the anchor's 95% interval, annotated with the width ratio. (b) The category prior under row- and instance-grouped folds. (c) A failure-predicting forest, identical but for whether targets are held out. All of it is a property of the protocol, not of any estimator.

A third component completes the picture. A natural companion to per-sample source selection is per-sample *abstention*: if a selector cannot choose the right source, perhaps a classifier can at least flag the predictions that will fail, and a system could decline to answer. We fit exactly that model, a random forest over 18 interpretable mask, depth, and scale cues, with failure defined as an ARE above one, to the corpus of $75{,}518$ uniform-protocol predictions, and score it two ways that differ only in the folds. Failures occur across all datasets, and the instance-grouped forest detects them: area under the ROC curve (AUROC) 0.93 on ECUSTFD (70 targets), 0.93 on MetaFood3D, 0.93 on NOCS, 0.84 on YCB (40 targets) and 0.83 on MADIMA23, against 0.98 for the row-shuffled control, which recognises objects rather than failures; MRGBD carries a single failing prediction in 764 and is too few to fit a curve on. Abstention has operational value here: keeping the 80% of predictions the forest judges safest cuts the pooled failure rate from 21% to 7%, and on ECUSTFD, whose one catastrophic item drives much of Sec. 5.2, from 33% to 18%. Four of the five datasets supporting the result pass TARGETS, so the claim is admissible under our own criteria, though it is exploratory (Sec. C.6).

We therefore require that any monocular-measurement benchmark that fits a prior, trains a selector, or reports confidence intervals be grouped by target instance. The same argument applies to the bootstrap (Sec. 5.1).

E.6 The effect of fold grouping

Table 15 gives the numbers behind Sec. E.5. Grouping by row rather than by physical target lets a target appear in both the training and test sets. When a per-class prior can be learned across instances, this overstates it by $1.8\times$ on ECUSTFD, $1.9\times$ on NOCS and $4.4\times$ on MRGBD. MADIMA23 is a

different case, and we separate it: with two or three rows per target, row grouping lets the prior read the target's own value from its other row, whereas instance grouping collapses the per-class prior onto the global median, so the $5.4\times$ ratio measures that degeneracy rather than a learnable prior. The nested-selected rule on the MRGBD backbone pool and the plain mean under the uniform scale rule capture 78% of the headroom under either grouping, so under the current protocol the grouping does not move it (Table 15). The sign change that motivated the check, from $+74\%$ to -32%, occurs under the fitted per-cell constants of Sec. D.6 with a learned selector chosen, a protocol no result in this paper uses, so it carries no evidential weight here.

Table 15: **Effect of fold grouping.** Row-level grouping lets a physical target inform its own prediction; instance-level grouping is used throughout the paper. *Degenerate: each MADIMA23 target appears in only two rows, so under row grouping the prior reads its volume from the other row.

Dataset	Row grouping	Instance grouping	Factor
Category prior ARE			
ECUSTFD	0.090	0.159	$1.8\times$
MRGBD	0.027	0.117	$4.4\times$
MADIMA23	0.081	0.436	$5.4\times^{*}$
NOCS	0.273	0.521	$1.9\times$
Nested-rule headroom			
MRGBD (backbone)	78%	78%	none

E.7 Robustness to the fold count

Because two headline numbers depend on how folds are grouped (Sec. E.5), the fold protocol itself warrants an audit. Table 16 covers it. The geometric chain does not depend on the folds, since it fits nothing; the category prior does. Its ECUSTFD error varies between 0.143 and 0.159 across 3, 5, and 10 folds, reminding us that a fitted baseline carries protocol variance that an unfitted method does not. The conclusion does not change: on every dataset, at every fold count, the same side wins. Five folds, fixed before these runs and used throughout, is the least favourable of the three settings for the ECUSTFD prior, so the reported comparison is conservative with respect to fold count. The comparison in Sec. 5.2 is a property of the data rather than of the protocol; its magnitude is not.

Table 16: **Robustness to the fold count.** Mean ARE↓ of the unfitted geometric chain and of the category prior refitted at 3, 5 and 10 grouped folds. Bold: the prior beats the chain. Sec. E.7 reads the sweep.

Dataset	Geometric chain	Category prior at k folds			More accurate
		k=3	k=5	k=10	
ECUSTFD	6.493	**0.143**	**0.159**	**0.150**	prior
MADIMA23	0.611	**0.440**	**0.436**	**0.423**	prior
MRGBD	0.350	**0.149**	**0.117**	**0.112**	prior
MetaFood3D	1.597	1.637	1.673	1.643	chain
YCB	17.224	**1.899**	**1.854**	**1.847**	prior
NOCS	3.287	**0.480**	**0.521**	**0.530**	prior

The fold *count* is not the only protocol choice. Grouped k-fold is deterministic once the targets are fixed, so the reported prior has no seed to vary; what can vary is which targets land together. Table 17 randomises that assignment over 50 draws at five folds. The prior varies by a standard deviation of 0.014 to 0.11 across datasets, and the same side wins across the entire 5–95 percentile range on every dataset, so the comparison does not rest on a fortunate partition.

Table 17: **Partition variance of the fitted prior.** The assignment of targets to five folds, randomised over 50 draws; the chain is unfitted and does not move. *Winner*: which side is lower across the whole 5–95 percentile range – partition variance, not a significance test (paired intervals in Table 18).

Dataset	Geometric chain	Prior reported	Mean	SD	5–95 pct.	Winner
ECUSTFD	6.493	0.159	0.180	0.033	[0.14, 0.24]	prior
MADIMA23	0.611	0.436	0.446	0.016	[0.42, 0.48]	prior
MRGBD	0.350	0.117	0.118	0.014	[0.10, 0.15]	prior
MetaFood3D	1.597	1.673	1.683	0.065	[1.62, 1.82]	chain
YCB	17.224	1.854	1.858	0.107	[1.73, 2.05]	prior
NOCS	3.287	0.521	0.537	0.068	[0.44, 0.61]	prior

E.8 PROTOCOL VARIANCE IN THE SEMANTIC COMPARISON

Two properties of the protocol behind Sec. 5.2: how far the fitted side moves with the fold partition, and what a row-level grouping would do to the same comparison.

Only the prior side of the comparison is fitted, so only it carries protocol variance. Sweeping the fold count moves the ECUSTFD prior between 0.143 and 0.159 (App. E.7); randomising which targets share a fold, at five folds, yields a mean of 0.180 over 50 draws, so the reported partition sits at the favourable end of that range, with a 5–95 percentile range of $[0.142, 0.245]$. The partition can be excluded from the comparison altogether on the one dataset that ships its own training split. Fitting the ECUSTFD category prior on the 83 trainval targets and merely scoring it on the 837 test items, with no folds drawn inside the evaluation split and therefore no partition variance, gives 0.195, between the cross-validated 0.159 and the partition mean 0.180; the chain still loses to it by $+6.298$, with a corrected interval of $[+0.143, +24.683]$. The confirmatory verdict of Sec. 5.2 therefore does not depend on where the prior is fitted. The reported value is the deterministic grouped partition fixed in advance; it sits at the favourable end of the seed range, and the prior is more accurate than the chain on every drawn partition, on every dataset where it wins at five folds.

Under row-level grouping, the natural default since each row is one evaluated image, the same physical target appears in both folds through its other views, distances or meals, and the prior memorises individual items: it then reports 0.090 on ECUSTFD and 0.027 on MRGBD, overstating its accuracy by ratios of 1.8 and 4.4, and by 1.9 on NOCS, where the prior beats the chain under either grouping. On MADIMA23, where each target appears in only two rows, the row-grouped prior reads the target's own volume from its other row (0.081), so the comparison there measures a degeneracy rather than a learnable prior (Sec. E.5). Any benchmark that compares against a category or dataset prior must group folds by instance, or it will overstate the prior and thereby understate the method; at three, five and ten grouped folds, the same side wins on every dataset (App. E.7).

A second concerns the paper's own control. Table 51 reports each pairing on the items its own two sources share, which differ between columns, so the constant-versus-prior comparisons in that table are not like-for-like. The released `robustness_checks.csv` rebuilds all four pairings on one item set per dataset, the intersection of the chain and the incumbent model, and reports every contrast with an instance-clustered paired interval under mean ARE and under the symmetric log error. Sec. 5.4 quotes those numbers; they are what the control rests on, and Table 51 is the per-pairing view.

A third property concerns the metric rather than the protocol. Mean ARE is asymmetric, and four of the six datasets carry a chain or a prior scoring above 1.0, the ARE of answering zero, so each verdict is repeated on the symmetric log error $|\log(\hat{V}/V)|$, with predictions floored at $1\,\mathrm{cm}^3$ as in the main text and the same instance-clustered resample. The prior is the more accurate of the two on all six: $+0.323$ $[+0.256, +0.386]$ on ECUSTFD, $+0.707$ $[+0.419, +0.997]$ on MADIMA23, $+0.171$ $[+0.059, +0.220]$ on MRGBD, $+0.438$ $[+0.117, +0.755]$ on MetaFood3D, $+0.455$ $[+0.103, +0.791]$ on YCB and $+0.383$ $[+0.146, +0.585]$ on NOCS, every interval excluding zero before correction. Two readings follow. On five datasets, the symmetric metric agrees with mean ARE, and the margins are not an artefact of the asymmetry. On MetaFood3D it disagrees, which is why Sec. 5.1 reports that dataset's SKILL verdict as metric-dependent; the TARGETS count there admits the interval, so the disagreement is about the metric and not about the power.

F THE SEMANTIC CHANNEL AGAINST THE CHAIN

This appendix compares the class prior with the full chain, its sensitivity analyses, the label-parity rerun, where geometry is most confident, the propagation to dietary energy, and the human reference.

F.1 THE COMPARISON IN FULL

Table 18 and Fig. 2a set the class prior against the chain. We fix the reference chain per dataset before the study: ZoeDepth with the hull operator on ECUSTFD, MetaFood3D, YCB, and NOCS; Depth Pro on MADIMA23; and MoGe-2 on MRGBD. It is the weakest backbone on ECUSTFD and NOCS and the strongest on MADIMA23 and MetaFood3D, so no reading rests on it: Table 19 repeats the comparison for all four backbones on all six datasets. The prior is the more accurate side in 23 of the 24 cells and in all 24 under the symmetric log error. At the nominal 95% level,

Table 18: **A prior with no image geometry against the geometric chain.** Mean ARE of the reference chain of Sec. 5.2, the best of the four backbones on the backbone pool's items, and priors fitted on training folds grouped by target, with target-clustered 95% intervals beneath. *Class prior*: keyed on the dataset's label; *Prior, VLM label*: the same table queried with the name the incumbent VLM returns, and *Recog.* how often that name matches; *Class-calibrated*: the chain rescaled by a per-class factor. η^2 is the in-sample variance the category explains. Last column: chain minus class prior, positive where the prior wins; **bold**, its interval excludes zero on a dataset passing TARGETS; [§], it fails. In the ARE columns bold marks the more accurate of chain and class prior.

				Mean ARE↓							
Dataset	**Cls.**	**Within CV**	η^2	**Ref. chain**	**Best backbone**	**Global prior**	**Class prior**	**Prior, VLM label**	**Recog.**	**Class-calibrated**	**Chain−prior**
ECUSTFD	19	0.098	0.94	6.493 [0.38,20.06]	1.417	0.659 [0.51,0.84]	**0.159** [0.12,0.21]	0.346	62%	11.006	**6.334** [0.21,19.91]
MADIMA23	34	0.073	1.00	0.611 [0.55,0.67]	0.611	0.457 [0.35,0.59]	**0.436** [0.32,0.57]	0.871	25%	0.879	**0.175** [0.02,0.31]
MRGBD	5	0.053	0.92	0.350 [0.27,0.38]	0.162	0.249 [0.13,0.32]	**0.117** [0.08,0.17]	0.117	100%	0.118	0.233[§] [0.12,0.28]
MetaFood3D	34	–	1.00	**1.597** [0.69,2.91]	1.597	1.673 [0.76,2.87]	= *global*	2.492	32%	3.484	-0.076 [-1.09,1.21]
YCB	40	–	1.00	17.224 [5.84,33.54]	13.474	1.854 [0.78,3.42]	= *global*	2.133	43%	15.703	**15.370** [4.07,31.55]
NOCS	5	0.332	0.57	3.287 [0.87,7.34]	1.922	0.603 [0.45,0.78]	**0.521** [0.27,0.85]	0.555	86%	4.920	2.766[§] [0.26,6.83]

Table 19: **The prior beats every chain we have, not only the reference chain.** Mean ARE of the geometric chain under each depth backbone, against the category prior on the same items, with the paired difference $\mathrm{ARE}_{\text{chain}} - \mathrm{ARE}_{\text{prior}}$ (Δ; positive where the prior is more accurate, **bold** where its instance-clustered interval excludes zero on a dataset passing TARGETS, [§] where it fails). Underlined: the reference chain Table 18 reports. [†]: the cell clears SKILL. A backbone's failed outputs are kept and scored as the error tables score them, so every ECUSTFD cell uses all 837 items; **bold** is at the nominal 95% level. Under the symmetric log error the prior is ahead in every one of the twenty-four cells.

Dataset	**Prior**	**ZoeDepth**		**Depth Pro**		**UniDepthV2**		**MoGe-2**	
	ARE↓	ARE↓	Δ	ARE↓	Δ	ARE↓	Δ	ARE↓	Δ
ECUSTFD	0.159	6.493	**+6.33**	1.588	**+1.43**	5.117	**+4.96**	1.417	**+1.26**
MADIMA23	0.436	0.656	**+0.22**	0.611	**+0.18**	0.793	**+0.36**	0.619	**+0.18**
MRGBD[§]	0.117	0.381	+0.26	0.162	+0.05	0.889	+0.77	0.350	+0.23
MetaFood3D	1.673	1.597	-0.08[†]	13.287	**+11.61**	5.846	**+4.17**	19.914	**+18.24**
YCB	1.854	17.224	**+15.37**	17.516	**+15.66**	13.474	**+11.62**	26.507	**+24.65**
NOCS[§]	0.521	3.287	+2.77	2.020	+1.50	1.922	+1.40	2.460	+1.94

its interval excludes zero in 15 of the 16 cells on datasets that pass TARGETS, all but MetaFood3D under ZoeDepth. Against the best backbone, the prior's margin is $\times 8.9$ on ECUSTFD, $\times 1.4$ on MRGBD and MADIMA23, $\times 7.3$ on YCB and $\times 3.7$ on NOCS.

The primary family. The family, its metric and its correction were fixed before the paired analysis, though not pre-registered: every volume dataset that then passed TARGETS, ECUSTFD, MADIMA23 and MetaFood3D, each against its best learnable prior, scored on mean ARE at the Bonferroni level $0.05/3$ (Table 36). YCB passed TARGETS only after its rebuild and stays outside. ECUSTFD's percentile interval excludes zero ($+6.334$, $[+0.193, +24.706]$), MADIMA23's does not ($[-0.016, +0.335]$), and MetaFood3D is unresolved (-0.076). The ECUSTFD interval depends on the method. One item, a coin estimate that saturated its clip bound and turned a doughnut into more than a cubic metre, carries 96% of the ARE difference, so the bootstrap is bimodal on whether that target is drawn, and the cluster-robust t interval spans zero ($[-8.48, +21.14]$). The defect has a one-line repair, letting the coin cue abstain when it saturates its bound, which takes the chain's mean ARE from 6.49 to 0.43 (App. I.5). The target-level evidence does not depend on it either: the prior wins on 64 of the 70 targets (Monte Carlo sign-flip $p = 5 \times 10^{-6}$, the smallest 200,000 draws can give), and without the item the difference is $+0.266$ $[+0.178, +0.357]$ at the same level.

Under the symmetric log error, declared as a sensitivity analysis, all three family members survive $0.05/3$: ECUSTFD $+0.323$ $[+0.240, +0.401]$, MADIMA23 $+0.707$ $[+0.364, +1.068]$ and MetaFood3D $+0.438$ $[+0.048, +0.823]$. Pooled over the four datasets that pass TARGETS, the log difference is $+0.45$ (random effects, t-based interval $+0.15$ to $+0.75$). Over 50 random fold partitions, every point estimate keeps its sign, and on MetaFood3D the chain stays ahead on mean ARE in all 50. Outside the family, YCB's global prior beats the chain by $+15.37$ $[+4.07, +31.55]$ (exploratory). Sign-flip tests, exact at 15 and 9 targets, give NOCS a nominal verdict (13 of 15 targets; $p = 0.02$ on ARE, 0.006 on the log error) and leave MRGBD unresolved ($p = 0.14$ and 0.30).

MetaFood3D is the one reversal. The chain is ahead there on mean ARE (1.597 against 1.673), but both exceed 1.0, the ARE of answering zero, and ARE caps an underestimate at one while leaving an overestimate unbounded. Under the log error, the prior is ahead (0.820 against 1.258). We report MetaFood3D as the dataset whose verdict depends on the metric, not as the one geometry wins.

Who supplies the label. Every prior above is keyed on the dataset's own label, so it is a lookup under perfect recognition. Queried instead with the name the incumbent VLM returns (Table 18), the prior is unchanged on MRGBD, where recognition is perfect, and nearly so on NOCS (86% recognised). It degrades on ECUSTFD (62%) from 0.159 to 0.346, still an order of magnitude ahead of the chain, and falls behind the chain on MADIMA23 and MetaFood3D, where the model names the object a quarter to a third of the time. The segmenter is also prompted with a label, so we reran the ECUSTFD chain with the same returned name; the rerun without the change reproduces the released run bit for bit. The prompt changes 80 of 837 volumes and leaves the chain's mean unchanged (6.492). With both sides on the predicted name, the prior is ahead by $+0.210$ $[+0.140, +0.278]$ in log error, while under ARE the interval touches zero ($[-0.002, +19.698]$; Table 37). The primary claim is therefore about a prior given the label. On the benchmarks whose classes a model can name, the predicted label serves almost as well.

Not only our chain. We reimplemented four published pipelines and ran each where applicable (Table 20; every run, including those on too few targets to report, in Table 6).

The most relevant result is ECUSTFD's coin-reference method (Liang & Li, 2017), run on the dataset it was written for, with the coin it expects. It is sixteen times more accurate than our chain on the same 771 items (0.432 against 7.010) and still loses to the class prior by $+0.277$ $[+0.170, +0.401]$ over 62 targets. DepthCalorieCam (Ando et al., 2019) and the MADIMA23 pipeline (Abdur Rahman et al., 2023) lose on ECUSTFD by $+0.71$ and $+0.36$. On MetaFood3D, all four runs beat the prior, with two intervals excluding zero, consistent with MetaFood3D being the only benchmark whose classes contain no lookup information.

Scale or shape. Scale is where the chain's error enters, since the depth maps are accurate in shape and wrong in size (App. D.7), but outside MRGBD it enters item by item. Table 21 fits a multiplicative correction on training folds. One factor per dataset removes 66% of the log error on MRGBD, bringing the chain level in line with the prior (-0.02 $[-0.04, +0.03]$), and none on ECUSTFD, YCB, or NOCS. A factor per class removes 9% and 23% of the log error on ECUSTFD and NOCS while worsening mean ARE. The factor an item needs spreads by an interquartile range of 0.04 on

Table 20: **Published pipelines, reimplemented and run under this protocol.** Mean ARE of each pipeline on the items it can process, beside our reference chain and the class prior on the same items, and the prior's margin over the pipeline (paired, target-clustered 95% interval; **bold** excludes zero). The ECUSTFD coin-reference method is the dataset's own, run on the dataset it was written for. A run is reported here where the pipeline's assumptions hold on that dataset, which is the selection Table 6 lists in full with the runs it excludes and why. ‡, the pipeline returns an estimate at or beyond the error of answering zero, so its margin carries no verdict. §, fails TARGETS.

				Mean ARE↓			
Dataset	**Pipeline**	n	**Tar-gets**	**Pipeline**	**Our chain**	**Prior**	**Prior's margin [95% CI]**
ECUSTFD	ECUSTFD coin-reference	771	62	0.432	7.010	0.154	**+0.277** [+0.170, +0.401]
	MADiMa23 pipeline (RealSense)	837	70	0.515	6.493	0.159	**+0.356** [+0.284, +0.425]
	DepthCalorieCam	837	70	0.872	6.493	0.159	**+0.713** [+0.650, +0.762]
	MADiMa23 pipeline (ZoeDepth)‡	837	70	0.995	6.493	0.159	+0.836 [+0.781, +0.878]
	MRGBD point-cloud‡	837	70	1.000	6.493	0.159	+0.841 [+0.786, +0.883]
MRGBD§	MRGBD point-cloud	21	8	0.422	0.343	0.128	**+0.294** [+0.270, +0.325]
MetaFood3D	MADiMa23 pipeline (ZoeDepth)	34	34	0.750	1.597	1.673	-0.923 [-1.996, -0.056]
	MADiMa23 pipeline (RealSense)	34	34	0.804	1.597	1.673	-0.869 [-1.801, -0.102]
	MRGBD point-cloud	34	34	0.846	1.597	1.673	-0.827 [-1.888, +0.007]
	DepthCalorieCam‡	34	34	0.999	1.597	1.673	-0.674 [-1.876, +0.242]
NOCS§	DepthCalorieCam‡	13,371	15	0.999	3.287	0.521	+0.478 [+0.150, +0.733]
	MADiMa23 pipeline (ZoeDepth)‡	13,371	15	0.999	3.287	0.521	+0.478 [+0.150, +0.733]
	MRGBD point-cloud‡	13,364	15	1.291	3.285	0.521	+0.770 [+0.327, +1.206]

Table 21: **No single scale correction, per dataset or per class, repairs the chain everywhere.** A multiplicative correction fitted on training folds and scored out of fold: one factor for the dataset (*Scaled*), the share of the chain's log error it removes, the median factor an item needed and the interquartile spread of that factor in depth units. A tight spread means one number would have fixed the dataset; a wide one means every item needs its own, which no scale rule can supply; **bold** IQR, below 0.10. Last column: the repaired chain against the prior, with its instance-clustered 95% interval; **bold**, the interval excludes zero on a dataset passing TARGETS; §, the dataset fails TARGETS.

		Mean ARE↓				Factor needed		
Dataset	n	**Chain**	**Scaled**	**Prior**	**Error removed**	**Med.**	**IQR**	**Scaled − prior**
ECUSTFD	837	6.493	6.760	0.159	0%	1.01	0.25	**+6.60** [0.22, 20.74]
MADIMA23	82	0.611	0.837	0.436	35%	1.40	0.60	**+0.40** [0.16, 0.68]
MRGBD§	191	0.350	0.100	0.117	66%	0.91	**0.04**	-0.02 [-0.04, 0.03]
MetaFood3D	34	1.597	3.484	1.673	14%	1.33	0.74	+1.81 [-0.14, 4.48]
YCB	800	17.224	15.703	1.854	none	0.97	0.80	**+13.85** [3.63, 28.29]
NOCS§	13,371	3.287	2.337	0.521	none	0.91	0.44	+1.82 [0.11, 4.41]

MRGBD against 0.25–0.80 elsewhere. No global or per-class correction repairs an error that varies by item; on YCB the operator's small-object artefact adds to it (App. D.7).

F.2 Is the chain a straw man? The prior against the strongest chain we can build

The comparison of Sec. 5.2 invites one objection: the prior wins because the chain is weak. We therefore gave the chain every advantage available without a new network run and asked again, on the same folds and items (Table 22). The variants were fixed before the first run: the chain as released; the chain times one factor fitted on the training folds, a constant the benchmark's protocol withholds from every chain; the chain times one such factor per class; and, on ECUSTFD, the reference chain with the coin-clamp repair of App. I.5. The analysis is exploratory.

As released, the prior is ahead on the log error in all 24 dataset–backbone cells and on mean ARE in 23, which reproduces Sec. 5.2. With one fitted factor it is ahead in 23 on the log error and 21 on mean ARE, and with a factor per class in 23 and 22. Choosing, per dataset and in hindsight, the variant with the lowest log error on the evaluation items themselves still leaves the prior ahead on five of the six datasets, with intervals excluding zero on ECUSTFD, MADIMA23 and NOCS and spanning it on MetaFood3D and YCB. The exception is MRGBD, where one factor brings the chain level with the prior, as App. F.1 already reports. The repaired ECUSTFD chain, whose mean ARE is 0.430 against the prior's 0.159, loses on 65 of 70 targets. Calibration narrows the prior's lead and does not reverse it: what the chain lacks is not a constant.

Table 22: **The class prior against the strongest chain we can build.** Per dataset, the chain variant with the lowest mean absolute log error on the evaluation items themselves, chosen in hindsight over four backbones and three variants (as released, times one factor fitted on the training folds, times one such factor per class); last row, the ECUSTFD reference chain with the coin-clamp repair. Difference is opponent minus prior in log error with a target-clustered 95% interval (**bold** excludes zero), and the last column counts the targets on which the prior is the more accurate side. Over all 24 dataset–backbone cells the prior is ahead on the log error in 24 as released, 23 with one fitted factor and 23 with a factor per class, and on mean ARE in 23, 21 and 22. [§], fails TARGETS.

Dataset	Strongest chain	**Mean ARE**: chain, prior		**Log error**: chain, prior		**Difference [95% CI]**	**Prior wins**
ECUSTFD	MoGe-2, a factor per class	0.687	0.159	0.434	0.154	**+0.279** [+0.209, +0.354]	62/70
MADIMA23	Depth Pro, a factor per class	0.876	0.436	0.756	0.485	**+0.272** [+0.063, +0.482]	25/36
MRGBD[§]	MoGe-2, one fitted factor	0.100	0.117	0.099	0.121	-0.022 [-0.046, +0.029]	4/9
MetaFood3D	ZoeDepth, one fitted factor	3.484	1.673	1.080	0.820	+0.260 [-0.016, +0.565]	20/34
YCB	Depth Pro, one fitted factor	25.160	1.854	1.258	1.012	+0.246 [-0.082, +0.567]	24/40
NOCS[§]	ZoeDepth, a factor per class	4.920	0.521	0.627	0.431	**+0.196** [+0.020, +0.378]	11/15
ECUSTFD	ZoeDepth, coin repair	0.430	0.159	0.486	0.154	**+0.332** [+0.264, +0.398]	65/70

F.3 A learned opponent

The chains of Sec. 5.2 are geometric, and a natural objection is that a trained regressor would do better. Without new network runs the honest version of that opponent is a regressor trained on what the chain computes: its volume and the pre-integration cues the learned selector reads. We fitted gradient boosting on those features, on the paper's folds grouped by target, once without the label and once with the training-fold class prior added as a feature (Table 23). It is not an end-to-end image model. The opponent, its features, its settings and the comparison were fixed before the first run; the analysis is exploratory.

Training closes most of the gap and does not reverse it where a class can be looked up. On ECUSTFD the regressor without the label takes the chain's log error from 0.477 to 0.209, and the prior, at 0.154, stays ahead by $+0.054$ (95% CI $+0.003$ to $+0.102$). On MADIMA23, MRGBD and NOCS the two are not distinguishable, every interval spanning zero. Where every class holds one target the prior is only a global median, and there learned geometry wins: on YCB by -0.324 (-0.519 to -0.122), and on MetaFood3D by a difference whose interval spans zero. Adding the label as a feature changes little. The headline of Sec. 5.2 should therefore be read as it is stated: a prior with no image geometry *rivals* the chain, and beats it where categories carry the volume; a trained regressor on the chain's features draws level with the prior on most benchmarks, and none of them shows it clearly ahead where a class prior exists.

Table 23: **A learned opponent: a regressor trained on the chain's own features.** Gradient boosting on the reference chain's log volume and its cached pre-integration cues, fitted on the paper's folds grouped by target, without the label and with the training-fold class prior as a further feature. It is not an end-to-end image model. Mean absolute log error, and the difference opponent minus prior with a target-clustered 95% interval (**bold** excludes zero; positive, the prior is ahead). On MetaFood3D and YCB every class holds one target, so the prior is the global median. §, fails TARGETS.

Dataset	Targets	Prior	Chain	No label	minus prior [95% CI]	With label	minus prior [95% CI]
ECUSTFD	70	0.154	0.477	0.209	**+0.054** [+0.003, +0.102]	0.171	+0.017 [-0.004, +0.038]
MADIMA23	36	0.485	1.191	0.461	-0.023 [-0.163, +0.105]	0.453	-0.031 [-0.164, +0.097]
MRGBD§	9	0.121	0.292	0.136	+0.015 [-0.018, +0.040]	0.113	-0.007 [-0.022, +0.021]
MetaFood3D	34	0.820	1.258	0.765	-0.055 [-0.326, +0.218]	0.755	-0.064 [-0.350, +0.220]
YCB	40	1.012	1.467	0.688	**-0.324** [-0.519, -0.122]	0.727	**-0.285** [-0.471, -0.092]
NOCS§	15	0.431	0.814	0.551	+0.120 [-0.076, +0.306]	0.554	+0.123 [-0.134, +0.378]

A trained image model. The second opponent is end to end: a linear head on frozen DINOv2 features of the object's pixels, fitted on the same folds (Table 24). It is given no label and no metric scale, but it sees the object, so it can read both the category and the instance. The features, the head, the folds and the comparison were fixed before the first run, and one deviation was made before any result on the datasets concerned: MADIMA23, MRGBD and MetaFood3D carry no crop boxes because the panel was shown the whole frame, so the head is shown the whole frame too. The analysis is exploratory.

The head is far more accurate than the zero-shot chain on five of the six datasets, and it lands on the prior. On ECUSTFD its log error is 0.172 against the prior's 0.154, a difference whose interval spans zero ($+0.018$, -0.014 to $+0.046$), and MADIMA23, MRGBD and NOCS are ties as well. On MetaFood3D, with one item per class, the prior stays ahead ($+0.208$, $+0.080$ to $+0.352$); on YCB, where the prior is only a global median, the head wins (-0.228, -0.425 to -0.034). A trained model that sees the pixels therefore reaches the accuracy of a lookup by name wherever there is a name to look up, and clearly exceeds it nowhere that a name exists. That is the discriminant-validity problem of Sec. 7 seen from the other side: on these benchmarks an image model's score does not say whether it measured the object or recognised it.

Table 24: **A trained image model as the opponent.** A linear head on frozen DINOv2 features of the object's pixels, fitted on the paper's folds grouped by target; it sees no label and has no metric scale. Mean absolute log error on the items that carry both a reference-chain estimate and an image (MADIMA23: the 40 frames the panel scored; NOCS: at most 60 rows per target); the difference head minus prior with a target-clustered 95% interval (**bold** excludes zero; positive, the prior is ahead); targets on which the prior is the more accurate side; and the head with the training-fold class prior added as a feature. On MetaFood3D and YCB the prior is the global median. §, fails TARGETS.

Dataset	Items	Targets	Prior	Chain	Image head	Head − prior [95% CI]	Prior wins	Head + label
ECUSTFD	837	70	0.154	0.477	0.172	+0.018 [-0.014, +0.046]	43/70	0.169
MADIMA23	40	34	0.491	0.825	0.453	-0.038 [-0.131, +0.058]	12/34	0.452
MRGBD§	191	9	0.121	0.292	0.097	-0.023 [-0.047, +0.032]	7/9	0.097
MetaFood3D	34	34	0.820	1.258	1.028	**+0.208** [+0.080, +0.352]	24/34	1.113
YCB	800	40	1.012	1.467	0.784	**-0.228** [-0.425, -0.034]	15/40	0.775
NOCS§	900	15	0.410	0.803	0.345	-0.065 [-0.185, +0.042]	6/15	0.347

F.4 THE PRIOR ON PREDICTED LABELS

Sec. 5.2 reports the class prior as a lookup, which assumes the object has already been recognised. The fairer question is whether the finding survives a label a method could produce itself. We fit a multinomial logistic head on the frozen DINOv2 features of App. F.3, on the same target-grouped training folds, predict each held-out item's class, and rebuild the prior from that prediction. The fallback is the paper's own: where the predicted class holds fewer than two training targets, the prior is the training-fold global median, which is what the prior does on MetaFood3D and YCB whatever the label. Nothing else changes, and the plan was fixed before the first run.

The finding survives, with a narrower margin where the label has to be predicted (Table 25). On ECUSTFD the head recovers 96% of the labels and the prior still beats the chain by 0.252 in log

Table 25: **The class prior when the label is predicted, not looked up.** A multinomial logistic head on the same frozen DINOv2 features, fitted on the paper's target-grouped training folds, predicts each held-out item's class; the prior is then the training-fold median of the items whose true class is that prediction. *Acc.* is out-of-fold label accuracy, which is 0 on the three datasets whose classes hold one target each, or all but two of them, so a held-out class is not in training. Mean absolute log error, and the chain minus each prior with a target-clustered 95% interval (**bold** excludes zero; positive, the prior is ahead).

			Mean abs. log error			Chain − prior		
Dataset	**Targets**	**Acc.**	**Chain**	**True**	**Pred.**	**True**	**Predicted**	**Targets**
ECUSTFD	70	0.96	0.477	0.154	0.226	**+0.323**	**+0.252** [+0.180, +0.322]	58/70
MADIMA23	34	0.00	0.825	0.491	0.519	**+0.334**	+0.307 [-0.000, +0.633]	19/34
MRGBD	9	0.95	0.292	0.121	0.159	**+0.171**	+0.133 [-0.017, +0.213]	5/9
MetaFood3D	34	0.00	1.258	0.820	0.820	**+0.438**	**+0.438** [+0.117, +0.755]	25/34
YCB	40	0.00	1.467	1.012	1.012	**+0.455**	**+0.455** [+0.103, +0.791]	27/40
NOCS	15	0.81	0.803	0.410	0.371	**+0.394**	**+0.432** [+0.281, +0.588]	14/15

error against 0.323 with true labels, on 58 of 70 targets. On NOCS (81%) the predicted-label prior is the better of the two, $+0.432$ against $+0.394$, because its coarser groups are also its more populated ones. On MetaFood3D and YCB every class holds one object, so the prior is the global median and a label cannot change it. The two places where the evidence weakens are MADIMA23, whose interval now touches zero ($+0.307$, $[-0.000, +0.633]$), and MRGBD, whose interval crosses it ($+0.133$, $[-0.017, +0.213]$) on nine targets. Both are datasets the validity criteria already mark as unable to carry a significance claim, and the two that do carry one, ECUSTFD on TARGETS and NOCS after its union, keep the result. The honest reading is that the prior's advantage is not an artefact of oracle labels, and that it is measured with least confidence exactly where the benchmark is thinnest.

F.5 THE THREE CHECKS OUTSIDE VOLUME ESTIMATION

Sec. 6 argues that the lessons should read the same way wherever a strong prior sits beside a weak measurement of the same quantity. We ran them once on a problem with no image, depth map or volume, with the released code unchanged: the body mass of the 333 Palmer penguins with every field recorded (Gorman et al., 2014; Horst et al., 2020). The class is species $\times$ sex, the entry a field guide gives, and the two sources are morphometric size indices, flipper length cubed and bill length times bill depth squared, neither fitted to mass. Their level is arbitrary, which the class-anchored rule ignores, because it fuses deviations. The data, the class, the sources and the quantities reported were fixed in the script before its first run, and the analysis is exploratory.

The problem is in the paper's regime (Table 26). The class explains 84% of the log-mass variance, against 83% on ECUSTFD, and either index alone, given one training-fold factor, is less accurate than the prior. Both pass the fusion condition of Eq. (4). Keeping the prior and adding an index's within-class deviation lowers the mean absolute log error from 0.0610 to 0.0565 with the flipper index, a gain of $+0.0045$ (95% CI $+0.0018$ to $+0.0071$), about 7%, and to 0.0566 with the bill index; both win all 30 partitions, and both permutation p values sit at the floor of $10{,}000$ draws. The measurement model predicts log-RMSE ratios of 0.930 and 0.937, and 0.929 and 0.940 are realised. A constant source returns the prior exactly.

The comparison with the combiners that nest the rule repeats what the held-out dishes showed (Sec. 5.6). With about 55 targets a class, the rule beats the log-space stack and gradient boosting for both indices, but it only ties the random-intercept model, and it ties the Mundlak regression for the flipper index and beats it narrowly for the bill index: where classes hold many targets, a combiner can afford its free parameters. One dataset is an illustration of the argument, not a demonstration of its generality.

A learned regressor beside a group prior. The second case is a standard tabular benchmark, the $21{,}613$ house sales of King County, and the setting is domain shift. The class is the zipcode and the prior its median price on the training folds. The source is a gradient-boosted regressor that has never seen the zipcode it predicts: it is trained on the other zipcodes only, on the house's own attributes, with zipcode, latitude and longitude withheld, so it ranks houses within a zipcode and is miscalibrated across them. The data, the source, the three sample sizes and the draws were fixed before the first run; the analysis is exploratory.

Table 26: **The class-anchored rule outside volume estimation: body mass of 333 Palmer penguins.** Classes are species × sex; they explain 84% of the log-mass variance ($\sigma_w = 0.077$; the prior's out-of-fold log RMSE is 0.077), and $n_{\text{eff}} = 333$. Top, columns as in Table 2: mean absolute log error of the prior and of the fused estimate, the gain with a target-clustered 95% interval, the one-sided within-class permutation p at 10,000 draws, random partitions won, and the log-RMSE ratio predicted from r_{wc} against the one realised. Bottom: the mean absolute log error of the rule, of the source alone with one training-fold factor, and of the combiners of Table 65 on the same folds; *, worse than the rule with an interval excluding zero. A constant source returns the prior exactly.

Source	r_{wc}	Eq. 4	Prior	Fused	Gain [95% CI]	Perm. p	Partitions	Ratio pred. / real.
flipper length^3	0.367	yes	0.0610	0.0565	**+0.0045** [+0.0018, +0.0071]	≤0.0001	30/30	0.930 / 0.929
bill length × depth^2	0.348	yes	0.0610	0.0566	**+0.0043** [+0.0018, +0.0069]	≤0.0001	30/30	0.937 / 0.940

Source	Rule	Source alone	Log stack	Mundlak	Random intercept	Boosting
flipper length^3	0.0565	0.0846*	0.0580*	0.0569	0.0565	0.0595*
bill length × depth^2	0.0566	0.2725*	0.0607*	0.0573*	0.0568	0.0616*

This is the reverse of the paper's regime: the zipcode explains 53% of the log-price variance and the source's within-class correlation is 0.82, a strong source beside a weaker prior (Table 27). The measurement model still predicts the outcome, a log-RMSE ratio of 0.577 against 0.580 realised, and the rule lowers the prior's mean log error from 0.271 to 0.158 and beats both the source with one fitted factor and the log stack. But it is not the best combiner here. With all sales the Mundlak regression, the random-intercept model and gradient boosting are better, each with an interval excluding zero, and the constraint buys no protection when targets are few: at three sales per zipcode the stack, the Mundlak regression and the random-intercept model beat the rule in all 20 draws. Anchoring on the prior's level pays when the prior is the strong side; when a zipcode's median rests on three sales and the source is strong, a combiner free to discount the prior does better. The rule is for the asymmetric case the paper studies, and this case marks where it ends.

Table 27: **A learned regressor beside a group prior: King County house sales.** Mean absolute log error of the zipcode prior, of the class-anchored rule, of the source alone with one training-fold factor and of the combiners of Table 65, with the log-RMSE ratio the measurement model predicts against the one realised. With all sales, * is worse and † better than the rule with an interval excluding zero. The subsampled rows are means over 20 draws, with in parentheses the draws in which that combiner beats the rule.

Sample	n_{eff}	r_{wc}	Ratio pred. / real.	Prior	Rule	Source	Log stack	Mundlak	Random int.	Boosting
all sales	21,613	0.817	0.577 / 0.580	0.271	0.158	0.231*	0.161*	0.156†	0.152†	0.152†
10 per zipcode	700	0.801	0.599 / 0.595	0.299	0.179	0.233 (0)	0.173 (18)	0.174 (19)	0.163 (20)	0.180 (10)
3 per zipcode	210	0.773	0.633 / 0.600	0.348	0.206	0.236 (0)	0.193 (20)	0.198 (20)	0.190 (20)	0.221 (2)

F.6 The gain in cubic centimetres

A gain of 0.016 in mean absolute log error is hard to act on, so Table 28 states the same result in the unit the application uses. On ECUSTFD the rule saves about 2 cm^3 of mean absolute error on a median object of 155 cm^3, taking the prior's 23.3 cm^3 to 21.2; on MADIMA23 it saves 12.9 cm^3 of 73.9, and on NOCS 8.7 cm^3 of 210.1. On MetaFood3D it costs 3.7 cm^3. Read in grams of food, a 2 cm^3 saving on a 155 cm^3 portion is far below the tolerance any dietary use would need, and we make no claim that it reaches one. The result the paper rests on is about what a benchmark can attribute, not about a deployable improvement.

F.7 Why the random-intercept model loses

Sec. 5.5 reports that a random-intercept model with empirical-Bayes class effects does worse than the class prior itself, which is surprising because that model is the textbook version of what the rule does by hand. Table 29 takes it apart. The model fits $\log V = a + b \log \hat{V}^{(s)} + u_c$ on target-level rows, so it estimates one intercept per class *jointly* with a slope on a badly calibrated source: on ECUSTFD that is 21 parameters from 70 target rows, against the rule's single slope with the class centre taken from the prior rather than estimated. Two checks separate the candidate explanations. Shrinkage is not the cause where classes are populated: on ECUSTFD the mean empirical-Bayes weight is 0.96, and refitting with no shrinkage at all changes the out-of-fold log error from 0.170 to

Table 28: **The class-anchored gain in the unit the application uses.** Mean absolute error of the class prior and of the rule on the reference cell of each volume dataset, in cm^3, beside the same errors in log space. *Median* is the median ground-truth volume, so the saving can be read against the size of the object.

			Mean abs. error (cm^3)			Mean abs. log error	
Dataset	**Items**	**Median**	**Prior**	**Rule**	**Saved**	**Prior**	**Rule**
ECUSTFD	837	155	23.3	21.2	+2.1	0.154	0.138
MADIMA23	82	105	73.9	61.0	+12.9	0.485	0.396
MRGBD	191	120	12.9	11.4	+1.5	0.121	0.107
MetaFood3D	34	91	86.4	90.1	-3.7	0.820	0.819
YCB	800	190	351.3	348.5	+2.9	1.012	0.953
NOCS	13371	301	210.1	201.4	+8.7	0.431	0.416

0.164, still behind the prior's 0.154. Where each class holds one target, by contrast, the between-class variance is estimated as zero, the weight collapses to zero, and the model discards the class entirely: that is MetaFood3D and YCB. The complement is the useful part of the result. Where classes do hold several targets the model is competitive, beating the prior on NOCS (0.414 against 0.431) and on MRGBD (0.095 against 0.121), where it is the best of the three. So the ordering is not an implementation defect but the cost of estimating class effects from few targets per class, and it is the same shortage of targets per class that the validity criteria measure.

Table 29: **Why the random-intercept model loses.** Its fit on target-level rows, beside the prior and the class-anchored rule, in mean absolute log error out of fold. τ^2 is the estimated between-class variance and *EB* the mean empirical-Bayes weight on the class effects; *unshrunk* repeats the model with that weight set to one.

			Targets			Mean abs. log error			
Dataset	**Targets**	**Classes**	**per class**	τ^2	**EB**	**Prior**	**Rule**	**RI**	**Unshrunk**
ECUSTFD	70	19	4	0.228	0.96	0.154	0.138	0.170	0.164
MADIMA23	36	34	1	0.284	0.95	0.485	0.396	0.447	0.455
MRGBD	9	5	2	0.019	0.69	0.121	0.107	0.095	0.102
MetaFood3D	34	34	1	0.000	0.00	0.820	0.819	0.849	0.849
YCB	40	40	1	0.000	0.00	1.012	0.953	0.991	0.991
NOCS	15	5	3	0.204	0.81	0.431	0.416	0.414	0.434

F.8 Nutrition5k with a class the image can supply

The dish class of Sec. 4 is the heaviest ingredient at 60% of the mass, computed from the published per-ingredient masses, so it is not available to a method at test time and the gain reported on Nutrition5k could be an artefact of it. We therefore predict the class from the image: a logistic head on frozen DINOv2 features of the same crops, fitted out of fold on the paper's target-grouped folds, and the class-anchored analysis repeated with its prediction in place of the paper's label.

The class is largely image-predictable, at 0.81 to 0.88 accuracy over 49 classes, and the result does not depend on the ground-truth definition (Table 30). Across the 21 panel runs the mean gain in ARE is 0.125 under the paper's class and 0.132 under the predicted one, and 20 of the 21 keep a positive gain; the incumbent's gain is identical to three decimals (0.156). Where the predicted class differs from the paper's it tends to be a coarser grouping, which raises the prior's error a little and the fused error by about as much. The reported gain is therefore a property of the dish's visible identity rather than of the ingredient table.

F.9 The best published method, not ours

The reference chain is a configuration we fixed, so the comparison in Sec. 5.2 could be read as a prior beating a straw man of our own making. Table 31 replaces it with the strongest published pipeline each dataset has, chosen by mean ARE among the four reimplemented methods of Table 6, and asks the same question of it.

The answer splits the way the rest of the paper splits. Where the comparison is powered, ECUSTFD with 62 targets and MetaFood3D with 34, the prior beats the best published method

Table 30: **Nutrition5k with a class the image can supply.** The paper's dish class is the heaviest ingredient at 60% of the mass, read from the published ingredient masses. *Acc.* is the out-of-fold accuracy of a logistic head on frozen DINOv2 features predicting that class from the crop; the last two columns are the class-anchored gain in mean ARE under the paper's class and under the predicted one, for every panel member.

		Prior	Fused ARE		Gain	
Panel member	**Acc.**	**ARE**	**Paper**	**Predicted**	**Paper**	**Predicted**
qwen3vl235b	0.87	0.552	0.352	0.359	+0.200	+0.201
qwen3vl30ba3b	0.81	0.548	0.348	0.372	+0.199	+0.238
qwen3vl32b	0.81	0.548	0.370	0.397	+0.178	+0.213
llama32v90b	0.87	0.552	0.375	0.377	+0.177	+0.183
qwen3vl8b	0.87	0.552	0.380	0.389	+0.172	+0.171
qwen3vl32bnf4	0.81	0.548	0.376	0.405	+0.172	+0.205
qwen3vl8bnf4	0.81	0.548	0.379	0.410	+0.168	+0.201
qwen25vl7b	0.85	0.547	0.386	0.390	+0.161	+0.165
internvl35	0.87	0.552	0.392	0.401	+0.160	+0.159
qwen25vl7b	0.85	0.559	0.403	0.407	+0.156	+0.156
qwen3vl30ba3b	0.88	0.547	0.395	0.395	+0.152	+0.154
qwen3vl32b	0.87	0.541	0.394	0.387	+0.148	+0.160
internvl35	0.87	0.552	0.405	0.418	+0.147	+0.142
molmo2	0.87	0.552	0.444	0.450	+0.108	+0.110
idefics3	0.87	0.552	0.455	0.468	+0.097	+0.092
idefics3	0.87	0.552	0.475	0.483	+0.077	+0.077
llavaov7b	0.87	0.552	0.478	0.493	+0.074	+0.067
qwen25vl7b (CoT)	0.86	0.559	0.511	0.519	+0.048	+0.048
qwen2vl2b	0.87	0.535	0.511	0.515	+0.025	+0.019
llavaov7b	0.86	0.547	0.535	0.534	+0.012	+0.016
qwen2vl7b	0.87	0.552	0.554	0.560	-0.002	-0.000

on ECUSTFD by 0.277 in mean ARE ($[+0.170, +0.401]$) and loses to it on MetaFood3D by 0.923 ($[-1.996, -0.056]$). Where it is not powered the direction favours the prior, by 0.294 on MRGBD's eight targets and 0.478 on NOCS's fifteen. Two things follow. The first is that the paper's claim does not depend on our chain: on ECUSTFD the dataset's own coin-reference method is sixteen times more accurate than our chain and still loses to a lookup table. The second is the same boundary the predicted-label and criteria results draw: MetaFood3D, the one volume dataset that passes every applicable criterion, is where a published pipeline beats the prior, and it is the dataset whose classes hold one object each. MADIMA23 and YCB are absent because their external runs do not align with our ground truth, which Table 6 records.

Table 31: **The best published method on each dataset, beside our chain and the class prior.** Of the four reimplemented pipelines, the one with the lowest mean ARE on that dataset, scored on the items it covers. The last column is that method minus the prior in mean ARE with a target-clustered 95% interval (**bold** excludes zero; positive, the prior is ahead). §, fewer than 30 targets, so the sign is a direction and not a significance claim. MADIMA23 and YCB have no reportable run, their external ground truth not aligning with ours.

			Mean ARE			
Dataset	**Targets**	**Best published method**	**Method**	**Our chain**	**Prior**	**Method − prior**
ECUSTFD	62	ECUSTFD coin-reference	0.432	7.010	0.154	**+0.277** [+0.170, +0.401]
MRGBD§	8	MRGBD point-cloud	0.422	0.343	0.128	**+0.294** [+0.270, +0.325]
MetaFood3D	34	MADiMa23 pipeline (ZoeDepth)	0.750	1.597	1.673	**-0.923** [-1.996, -0.056]
NOCS§	15	DepthCalorieCam	0.999	3.287	0.521	**+0.478** [+0.150, +0.733]

F.10 Does the rule transfer, and what sets its size?

Two probes ask whether the class-anchored rule is a method or a per-dataset fit, and what decides how much it recovers. Both reuse the rule exactly as Sec. 5.5 defines it, changing one thing at a time.

The slope transfers. The rule's only free number is the within-class slope b. We fit it on the whole of one dataset and substitute it, unchanged, for another dataset's own out-of-fold slope (Table 32). A borrowed slope does about as well as a fitted one wherever the donor has within-class structure to fit: ECUSTFD's own rule reaches 0.134 in log error and NOCS's slope reaches 0.133 on the same items, MRGBD's 0.135. Where the recipient is too thin to estimate b for itself, borrowing is better than fitting: MADIMA23's own slope, 0.015 from 36 targets, recovers 0.006 of log error,

while ECUSTFD's slope recovers 0.063 and MRGBD's 0.073. The two datasets whose classes hold one object each fit $b = 0$ and so return the prior; a borrowed slope moves them too, by 0.035 on MetaFood3D and 0.074 on YCB, though there it is a global recalibration rather than a within-class one, since their anchor is the global median. The practical reading is that a benchmark too small to fit the rule can still use it.

Coarsening the class widens the opening. The measurement model says the prior's strength and the rule's headroom are two views of one quantity. Merging a dataset's classes at random into k groups weakens the prior by a controlled amount, and the rule should recover more of what the label no longer explains. It does (Table 33): on ECUSTFD the prior's log error rises from 0.154 at its native 19 classes to about 0.50 at two, while the rule's gain rises from 0.021 to 0.083; on MADIMA23 from 0.006 to about 0.10, and on YCB from nothing to about 0.09. This is the clearest statement of the paper's mechanism: the gain is not a property of geometry alone but of how much the class label has left unexplained, which is what the SPREAD criterion measures.

Table 32: **The rule's slope transfers between datasets.** b is the within-class slope fitted on the whole of the donor dataset and then substituted, unchanged, for the recipient's own out-of-fold slope; everything else in the rule is the recipient's. Mean absolute log error out of fold (**bold**: at least as good as the recipient's own fit). MetaFood3D and YCB fit $b = 0$ because each of their classes holds one object, so their own rule returns the prior.

				Slope borrowed from					
Dataset	b	**Prior**	**Own rule**	**MADIMA23**	**MRGBD**	**MetaFood3D**	**YCB**	**NOCS**	**ECUSTFD**
ECUSTFD	0.145	0.154	0.134	0.150	0.135	0.154	0.154	**0.133**	–
MADIMA23	0.015	0.485	0.478	–	**0.412**	0.485	0.485	**0.427**	**0.421**
MRGBD	0.192	0.121	0.109	0.120	–	0.121	0.121	0.112	0.110
MetaFood3D	0.000	0.820	0.820	**0.814**	**0.787**	–	**0.820**	**0.786**	**0.784**
YCB	0.000	1.012	1.012	**1.003**	**0.937**	**1.012**	–	**0.955**	**0.947**
NOCS	0.121	0.431	0.416	0.428	**0.405**	0.431	0.431	–	**0.411**

Table 33: **Coarsening the class weakens the prior and widens the rule's opening.** Classes merged at random into k groups, 25 draws per k. Each cell is the prior's mean absolute log error and the rule's gain over it. The mechanism of Sec. 3.2 predicts the pattern: what the label stops explaining is what the source is left to explain.

	Classes	**Prior's log error / the rule's gain, at k classes**							
Dataset	**native**	$k = 40$	$k = 34$	$k = 19$	$k = 12$	$k = 8$	$k = 5$	$k = 3$	$k = 2$
ECUSTFD	19	–	–	0.154/+0.021	0.371/+0.033	0.430/+0.045	0.465/+0.063	0.501/+0.083	0.495/+0.083
MADIMA23	34	–	0.485/+0.006	–	0.588/+0.091	0.599/+0.102	0.576/+0.107	0.545/+0.101	0.511/+0.086
YCB	40	1.012/+0.000	–	–	1.240/+0.107	1.250/+0.091	1.149/+0.090	1.089/+0.080	1.044/+0.074
NOCS	5	–	–	–	–	–	0.431/+0.015	0.576/+0.009	0.574/-0.001
MetaFood3D	34	–	0.820/+0.000	–	1.011/+0.014	1.013/+0.017	0.937/+0.023	0.922/+0.014	0.854/+0.021
MRGBD	5	–	–	–	–	–	0.121/+0.012	0.188/+0.056	0.194/+0.064

F.11 How large would a benchmark have to be?

The validity criteria say which datasets cannot carry a claim. The question a dataset author actually faces is the other one: how many targets are enough. Eq. (4) gives a necessary condition, $r_{\mathrm{wc}}^2 > 1/n_{\mathrm{eff}}$; this turns it into a sample size. We simulate benchmarks from the model of Sec. 3.2, run the rule on the paper's folds and take the paper's target-clustered interval, over a grid of within-class correlation, number of targets and targets per class, 300 replicates per cell.

Table 34 reads as a design rule: with at least five targets per class, resolving a gain from a source whose within-class correlation is 0.5 takes about 150 targets, 0.4 takes 250 and 0.3 takes 500; below $r_{\mathrm{wc}} = 0.2$ no benchmark in the grid reaches 80% power, because the gain is then smaller than the cost of estimating the slope. Targets per class matter less than the total: five per class is nearly as good as ten, and two is materially worse.

Two readings follow, and the second is uncomfortable for this paper. The first is that the volume benchmarks are an order of magnitude too small for the effect they are being asked about: every geometry cell here has r_{wc} between 0.10 and 0.36 against 4 to 69 effective targets, where the table asks for 500 or more. The second is that our own confirmed ECUSTFD gain was obtained at that sample size, so it was measured at low power, and an effect detected at low power is more likely to overstate the population value than to understate it. The pre-registered replication on held-out objects is the guard against that, and it came back larger rather than smaller; on the volume datasets,

which carry no such replication, neither the size of a gain nor its sign is established at the nominal error rate. The simulation also shows how often a thin benchmark resolves anything at all: at $r_{\mathrm{wc}} = 0.1$, where the rule's mean effect is a small *loss* (-0.011 at 20 targets, -0.004 at 50), the target-clustered interval excludes zero in 17 to 22% of simulated benchmarks at 50 targets or fewer, in either direction. The grid holds no null cell, so this is a resolution rate and not a type-I rate, and it is one more reason the § rows report a direction and not a claim.

Table 34: **How large a benchmark has to be.** The smallest number of distinct targets at which the class-anchored gain is resolved in 80% of simulated benchmarks (the interval excludes zero in either direction), by the source's within-class correlation r_{wc} and the number of targets per class. Simulated from the model of Sec. 3.2 with the paper's folds and its target-clustered interval, 300 replicates per cell. Every geometry cell in this paper has r_{wc} between 0.10 and 0.36 and at most 69 effective targets.

Targets per class	r_{wc}						
	0.1	0.2	0.3	0.4	0.5	0.6	0.7
2	> 500	> 500	> 500	500	250	150	75
5	> 500	> 500	500	250	150	100	50
10	> 500	> 500	500	250	150	75	50

The simulation says what is provable. The refits of App. I.8 say what happens, on the two datasets large enough to ask: Table 35 gives the fraction of 18,638 real refits in which the rule beat the prior. The two tables answer different questions and should be read together. For the rule to help, a source with $r_{\mathrm{wc}} \geq 0.4$ and about 50 targets is enough, and it then helps in 99% of refits; below $r_{\mathrm{wc}} = 0.3$ it helps in only two thirds, however many targets are drawn. For the same gain to be *resolved*, Table 34 asks for several hundred targets. A benchmark can therefore be large enough to use the rule and far too small to prove it works, which is the position every volume dataset in this paper is in, and it is the clearest statement we can make of what the next dataset needs.

Table 35: **When the rule helps, on real data.** The fraction of the 18,638 refits of App. I.8 in which the class-anchored rule beats the prior, by the number of targets the refit drew and the source's within-class correlation (**bold**: at least 0.95). Helping is not the same as being provable: Table 34 asks for hundreds of targets before the same gain is resolved at 80% power.

Targets	r_{wc}					
	< 0.3	0.3–0.4	0.4–0.5	0.5–0.6	0.6–0.7	> 0.7
≤ 24	0.67	0.73	0.79	0.81	0.92	0.87
25–48	0.64	0.83	0.92	**0.99**	**1.00**	**1.00**
49–96	0.70	**0.96**	**0.99**	**1.00**	**1.00**	**1.00**
97–192	0.82	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
193–384	0.85	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
> 384	0.84	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

F.12 The design rule, turned back on this paper

If the sample size in Table 34 is worth anything, the cells it calls large enough should be the ones whose intervals exclude zero. They are. Of the 82 cells whose classes hold at least two targets, 26 meet the requirement at their own within-class correlation and 25 of those resolve; of the 56 that do not meet it, 11 resolve. Those 11 divide 6 to 5 between gains and harms, and 20% is the rate at which the simulation resolves anything at all on samples this thin. The six positive cells sit on three datasets, two of them share ECUSTFD's segmenter, scale rule and operator, and the MRGBD cells rest on nine targets, so they are directions, not a count of confirmed gains. Every one of the 26 is on Nutrition5k or OmniFood8K, which hold 1,175 and 7,952 effective targets. Not one volume cell qualifies. The largest carries 136 effective targets, at a within-class correlation the grid does not resolve at any size; of the volume cells whose correlation does carry a finite requirement, that requirement runs from 100 to 500 targets while the cells themselves hold 4 to 69. The paper's positive volume results are therefore the ones the rule says should be read as directions, and its powered results are the ones on mass, which is the sharpest way we can state what the field is missing: not a better pipeline, a volume dataset of the size the measurement model has been asking for all along.

F.13 THE CHAIN-VERSUS-PRIOR VERDICTS UNDER DECLARED SENSITIVITY ANALYSES

We fixed the primary family, metric, and correction per Sec. 5.2 but did not pre-register them, so we report five analyses alongside them that do not depend on those choices (Table 36). The cluster-robust (CR1) t interval shows why the percentile interval on ECUSTFD is wide: one item carries 96% of the ARE difference, and the bootstrap distribution is bimodal according to whether that target is drawn, so the CR1 interval spans zero while the percentile interval does not. The target-level tests don't care: the prior wins on 64 of 70 ECUSTFD targets. Sign-flip tests are valid at any number of targets, exact up to 20 and Monte Carlo above, where the observed statistic counts as one of the $200{,}000$ draws so that p is never zero; they also give NOCS a nominal verdict and leave MRGBD unresolved. Under the log error, every family member survives its correction. Across 50 random fold partitions, every point estimate keeps its sign, including MetaFood3D: the chain stays ahead on mean ARE in all 50. The log-error verdict is robust to the $1\,\mathrm{cm}^3$ floor applied to the seven failed chain outputs, which are exact zeros: the ECUSTFD log difference is 0.361, 0.342, 0.323, 0.310 and 0.304 at floors of 0.01, 0.1, 1, 5 and $10\,\mathrm{cm}^3$.

Table 36: **The chain-versus-prior verdicts under the declared sensitivity analyses.** Primary: mean ARE, percentile target bootstrap, Bonferroni at $0.05/3$ within the family fixed with the rule (ECUSTFD, MADIMA23, MetaFood3D); YCB, MRGBD and NOCS at 0.05 and outside it. Beside it: the cluster-robust (CR1) t interval at the same level, the symmetric log error at the same level, the targets on which the prior wins, the exact sign-flip p on per-target mean differences (valid at any number of targets), and the partitions of 50 in which the prior is ahead. Pooled log difference (DerSimonian–Laird): targets-passing: $+0.45$ [+0.27, +0.63]; all six: $+0.36$ [+0.23, +0.49]. [§], fails TARGETS.

Dataset	Family	α	ARE difference [corrected]	CR1-t interval	Log difference [corrected]	Prior wins targets	Sign-flip p (ARE)	Sign-flip p (log)	Partitions
ECUSTFD	yes	0.05/3	+6.334 [+0.193, +24.706]	[-8.476, +21.145]	+0.323 [+0.240, +0.401]	64/70	5×10^{-6}	5×10^{-6}	50/50
MADIMA23	yes	0.05/3	+0.175 [-0.016, +0.335]	[-0.007, +0.358]	+0.707 [+0.364, +1.068]	24/36	0.09	0.00017	50/50
MRGBD[§]	no	0.05	+0.233 [+0.116, +0.280]	[+0.162, +0.304]	+0.171 [+0.059, +0.220]	6/9	0.14	0.30	50/50
MetaFood3D	yes	0.05/3	-0.076 [-1.309, +1.536]	[-1.567, +1.415]	+0.438 [+0.048, +0.823]	22/34	0.93	0.01	0/50
YCB	no	0.05	+15.370 [+4.071, +31.550]	[+0.838, +29.903]	+0.455 [+0.103, +0.791]	30/40	0.00085	0.02	50/50
NOCS[§]	no	0.05	+2.766 [+0.262, +6.832]	[-1.134, +6.666]	+0.383 [+0.146, +0.585]	13/15	0.02	0.0056	50/50

F.14 LABEL PARITY

The chain's segmenter is prompted with the dataset's label, and so is the class prior, so the headline comparison gives both sides the same access to the category. Sec. 5.2 also queries the prior with the name the incumbent VLM returns; here the ECUSTFD chain is rerun with its segmenter prompted by that same name (`ONEVOL_LABEL_OVERRIDE` in the released code), so both sides must recognise. The run without the override reproduces the released run bit for bit (897 of 897 items), so the difference is the labels. The prompt changes 80 of the 837 test volumes and leaves the chain's mean unchanged (Table 37). GroundingDINO finds the one salient food on an ECUSTFD crop, whatever it is called, so the chain barely depends on the label, and the predicted-label comparison was already close to symmetric.

Table 37: **Label parity on ECUSTFD.** The chain's segmenter prompted with the dataset's label or with the name the incumbent VLM returns, against the class prior keyed the same way, on the 837 test items over 70 targets. Differences are chain minus prior (positive where the prior wins), with target-clustered 95% intervals; the third column drops the one degenerate chain item.

Label access	Chain	Prior	ARE difference	Without the degenerate item	Log difference
oracle label, both sides	6.493	0.159	+6.334 [+0.211, +19.905]	+0.266 [+0.194, +0.341]	+0.323 [+0.256, +0.386]
predicted label, both sides	6.492	0.346	+6.145 [-0.002, +19.698]	+0.078 [-0.027, +0.169]	+0.210 [+0.140, +0.278]
predicted label, prior only	6.493	0.346	+6.147 [-0.002, +19.699]	+0.078 [-0.027, +0.171]	+0.211 [+0.139, +0.280]

F.15 WHERE GEOMETRY IS MOST CONFIDENT

If the monocular chain failed only on a hard tail, it would remain useful on the rest, and the finding of Sec. 5.2 would be about coverage rather than about capability. We test this directly. A random forest is fitted on the pre-integration cues of Sec. H.7 to predict each item's relative error, under the same instance-grouped folds used everywhere else, so an item's score never depends on itself. We then rank items by predicted error and score both the chain and the category prior on the safest fraction, from a tenth to all of it.

Table 38 gives the result. The chain is more accurate than the prior in one of the thirty (dataset, coverage) cells, NOCS at half coverage. The tail the ranking is meant to remove is real: on YCB the chain's mean ARE falls from 17.22 over all 800 items to 1.49 over the 80 it judges safest, a factor of twelve. The prior on those same 80 items reads 1.12. So the chain's catastrophic aggregate is indeed carried by a minority of items, and removing them still does not put geometry ahead. The failure is one of capability, and abstention does not repair it.

Table 38: **Geometry does not win even where it is most confident.** Mean ARE of the chain on the safest fraction of items, ranked out of fold by a forest over the pre-integration cues of Sec. H.7; the prior is scored on the same items. **Bold**: the chain is more accurate than the prior there. The failure is one of capability rather than of coverage: on YCB the chain's error falls from 17.2 at full coverage to 1.5 at a tenth, and the prior is still ahead.

	Chain, mean ARE↓ at coverage						
Dataset	10%	25%	50%	75%	90%	100%	**Prior**
ECUSTFD	0.24	0.26	0.31	8.51	7.19	6.54	0.19
MADIMA23	0.43	0.43	0.47	0.53	0.58	0.61	0.44
MRGBD	0.30	0.31	0.33	0.35	0.35	0.35	0.12
YCB	2.84	1.95	1.49	1.62	2.58	17.22	1.85
NOCS	0.70	0.62	**0.60**	0.72	0.84	3.29	0.52

F.16 A HUMAN REFERENCE

Seventeen people estimated the volume of nineteen MetaFood3D scenes from the benchmark images and the food label, and every source was rescored on exactly those scenes (App. F.17). The crowd's mean ARE is 0.623. A bootstrap that resamples both participants and scenes yields a range of 0.30 to 1.10; scene by scene, the crowd beats our reference chain by 1.75 (95% CI 0.19–3.85) and the fused pool by 1.29 (0.17–2.79). It is not distinguishable from the two best reimplemented pipelines (-0.04, CI -0.53 to $+0.47$; -0.09, CI -0.57 to $+0.43$). Nineteen scenes fail TARGETS, so this is a reference point, not a verdict.

F.17 WHAT SEVENTEEN PEOPLE ACHIEVE ON THE SAME IMAGES

So far, every comparison has been between algorithms. The obvious missing reference is a person, and MetaFood3D provides one: seventeen participants estimated the volume of nineteen of its scenes from the same single images used for the benchmark scores. We rescore every source in this paper on exactly those nineteen scenes, using the same protocol as elsewhere, and report the results in Table 39 and Fig. 8. Nineteen scenes fail TARGETS, so no scene-clustered interval is admissible, and the table presents point estimates. One interval is available on the other axis: seventeen participants are themselves a resampling unit, and resampling them answers a question the scene axis cannot, namely how far the crowd figure would move if a different seventeen people had answered. Over 20,000 participant-clustered draws, the crowd's mean error is 0.623 $[0.399, 0.914]$. That interval excludes every source of our own on these scenes, the nearest being the fused pool at 1.917, and it contains the two best reimplemented pipelines, at 0.667 and 0.714. The crowd is therefore ahead of our chain by a margin that survives resampling the people, and is not distinguishable from the published pipelines: seventeen different respondents could plausibly have produced either ordering. We read the table as a calibration of what the task is worth attempting, not as a verdict about which pipeline is best.

How we chose the nineteen, and what that costs, is worth stating rather than leaving to inference. The form was built before any analysis in this paper and asked twenty questions covering nineteen distinct scenes, one of them asked twice under two spellings of its label; the scenes are numbers 1–16, 23, 24 and 25 of the challenge subset in its released order, taken by position rather than by a random draw, which is a convenience sample and not a designed one. What it is not is a favourable one: the nineteen span the full range of the thirty-four (5.5 to 828.2 cm^3, against a median of 87 where the full subset reads 91), and every source in the table is rescored on exactly them. The per-participant answers are released with the artefacts, so we can recompute the distribution behind every row here. The instrument also affords one reliability check, which we report rather than average away: the same question asked twice under two spellings drew two answers from each of the seventeen participants, and those pairs differ by a median factor of 1.33, correlating at $r = 0.83$ in log space. A person asked the same question twice, minutes apart, moves their own answer by a third, which bounds how finely we can read any of these individual numbers.

Three observations follow. First, the crowd is far ahead of our reference chain and not distinguishable from the best pipeline: the chain reaches 2.371 in these scenes, the crowd's mean estimate 0.623, a factor of 3.8 and its median estimate 0.527, and the best of the reimplemented published pipelines, the MADIMA23 pipeline in its ZoeDepth form, 0.667 (the MRGBD method 0.714; the RealSense form 0.977; DepthCalorieCam is a failed run at 0.999). The gap to our chain is a factor of four; the gap to the best pipeline is within the noise of nineteen scenes. A bootstrap that resamples both participants and scenes, and pairs every contrast scene by scene, puts the crowd's mean at $[0.30, 1.10]$ and its lead over our chain at 1.75 (95% CI 0.19–3.85) and over the fused pool at 1.29 (0.17–2.79), while its difference from the two best pipelines, -0.04 (-0.53 to $+0.47$) and -0.09 (-0.57 to $+0.43$), spans zero. A bootstrap over participants alone, $[0.40, 0.91]$, holds the nineteen scenes fixed and is too narrow. Second, aggregation matters: a single participant averages 0.849, the median participant reaches 0.659, and the seventeen range from 0.346 to 1.851. The median person with one glance beats every source of our own, and is not distinguishable from the best reimplemented pipeline, 0.659 against 0.667, a gap of one part in eighty that the next paragraph shows the metric decides. Third, the automatic sources bunch rather than spread: the cross-validated global prior reaches 1.978, nested rule selection over the full pool 1.917, and the chain 2.371, while a constant answer (3.644) and the VLM (6.738; 2.551 when given the scene's label, as the participants were) remain behind. Fusion helps, and the fused pool is the best of our own sources on these scenes, but it remains a factor of three behind the crowd and behind the two best reimplemented pipelines.

Two of these rows sit above 1.0, the ARE of answering zero, so Table 39 carries the symmetric log error beside ARE, and the reading survives it. The crowd's mean error is 0.440 versus the chain's 1.440, so the factor of 3.8 becomes 3.3, and the direction is unchanged. One ordering does move: the median participant reads 0.596 against the best reimplemented pipeline's 0.533, reversing their positions under mean ARE (0.659 against 0.667). The two are separated by less than the metric that separates them, so we draw no conclusion about which is better; both metrics agree only that a person with one glance is closer than any source we built. The one row the symmetric metric rewrites is DepthCalorieCam, whose 0.999 under ARE becomes 7.10: it is the unit failure Table 6 marks, not a method that ties the crowd, and mean ARE alone cannot say so.

The comparison completes the argument of Sec. 5.2 from the other side. MetaFood3D holds one target each class, so no category prior can be learned; the shortcut that dominates ECUSTFD and MRGBD is simply unavailable, and what remains is a task where geometry does real work – it is the one dataset on which the chain clears SKILL over the full item set, on a point difference whose interval spans zero, and it clears it on mean ARE alone: the symmetric log error puts the prior ahead by $+0.438$ $[+0.117, +0.755]$ (Sec. 5.1) – and where a small crowd, given the label, is as accurate as the best published pipeline and four times as accurate as our chain, while the median person is behind that pipeline. The paper's headline is not that geometry is unnecessary. It is that on benchmarks whose classes barely vary in volume, a category lookup answers the question without measuring anything, and those are most of the benchmarks in use. Where a benchmark denies that shortcut, as MetaFood3D does, the monocular chain is no longer beaten outright on mean ARE, though its lead there spans zero and reverses under the log error; the best of the published pipelines reaches a crowd's accuracy, and our own chain does not, which both endorses the geometry and states how much of the task remains for a single view of a single person.

G THE VISION–LANGUAGE CHANNEL

This appendix provides the full panel, the single-model study that preceded it, the prompt forms, the check on view multiplicity, and the geometric channel on a mass dataset.

G.1 THE FOURTEEN-MEMBER PANEL IN FULL

A negative result from one model is a fact about that model. The panel spans seven vision encoders, three language-model families and 2B to 235B parameters, scored on identical frozen crops (Table 40). Fourteen members are scored: twelve distinct models and two of them rerun in 4-bit, so a panel median weights those two twice. A *run* is one member on one dataset under one prompt form. Member-level counts use each member's direct prompt, and the class-supplied, chain-of-thought and metric-extents prompts are ablations (App. G.4). YCB has one object per class and enters only the pools and controls. Quantisation to 4 bits (NF4, the 4-bit NormalFloat format) changes ARE

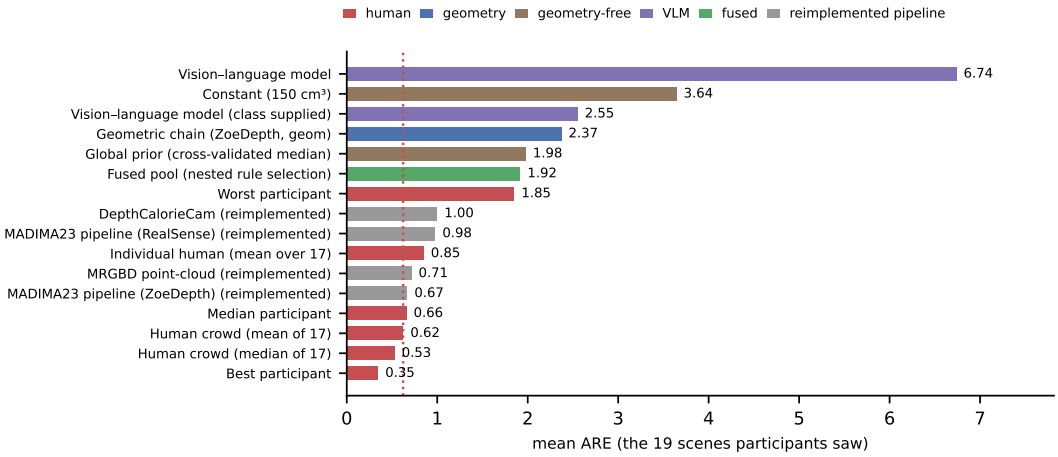

Figure 8: **Seventeen people against every source in the paper.** Mean ARE on the nineteen scenes shown to participants; every source is rescored on exactly those scenes; the dotted line is the crowd mean. The crowd mean is not distinguishable from the best reimplemented pipeline; the median participant beats our chain, the fused pool, and, narrowly, the best pipeline under mean ARE, and lags behind it under log error. Point estimates only: the scenes fail TARGETS.

Table 39: **Seventeen people on the MetaFood3D survey subset.** Mean ARE on the 19 scenes 17 participants estimated from the same single images the benchmark scores, with every computational source rescored on exactly those scenes. Δ: against the crowd mean; bold beats it. The symmetric log error is shown beside ARE because several rows exceed 1.0, the ARE of answering zero. †, a published pipeline reimplemented here. Point estimates only: the scenes fail TARGETS[§], so no interval over scenes is admissible; the participant-clustered interval on the crowd row is in Sec. F.17.

Source	Mean ARE↓	Log err.↓	Δ vs crowd
Best participant	**0.346**	0.41	-0.278
Median participant	0.659	0.60	+0.036
Worst participant	1.851	1.17	+1.228
Human crowd (mean of 17)	0.623	0.44	+0.000
Human crowd (median of 17)	**0.527**	0.42	-0.096
Individual human (mean of 17)	0.849	0.64	+0.226
Geometric chain (ZoeDepth, geom)	2.371	1.44	+1.748
Global prior (CV median)	1.978	0.95	+1.354
Constant (150 cm^3)	3.644	1.08	+3.020
Vision–language model	6.738	1.41	+6.115
Vision–language model (class supplied)	2.551	0.93	+1.927
DepthCalorieCam†	0.999	7.10	+0.375
MADIMA23 pipeline (RealSense)†	0.977	0.97	+0.354
MADIMA23 pipeline (ZoeDepth)†	0.667	0.53	+0.044
MRGBD point-cloud†	0.714	0.55	+0.090
Fused pool (nested rule selection)	1.917	0.96	+1.294

by -0.05 to $+0.13$ and within-class correlation by at most 0.04 on the three datasets the panel's conclusions rest on.

On ECUSTFD, scored per row, within-class correlation ranges from 0.02 to 0.11 across the panel and never reaches 0.2, so no member explains more than 3% of the within-class variance. Per row understates a model whose answers vary across views of one object. With each target aggregated to one row, the median member reaches $r = 0.22$, and four of the thirteen ECUSTFD members have intervals excluding zero, at $r = 0.27$–0.36 (Fig. 9; App. G.5). None beats the class prior alone, and none gains significantly over it in the class-anchored combination.

On plated portions, the picture inverts. On Nutrition5k, the within-class correlation is 0.65–0.73 for six of the eleven members run at 16-bit precision (bf16). The most accurate member, Llama-3.2-90B, reaches an ARE of 0.44 against the class prior's 0.55. Qwen2-VL-7B emits eleven distinct values across $1{,}347$ dishes and scores zero, a failure it repeats on OmniFood8K. There the median member's within-class correlation is 0.74, thirteen of fourteen intervals exclude zero, and the best member reaches 0.26 against the prior's 0.28. Calibration, not information, is what the members lack: median bias ranges from $\times 0.67$ to $\times 1.50$, so the panel median (0.59 on Nutrition5k) trails the best member (0.44), and the per-item oracle reaches 0.16. A source that ranks but does not calibrate is the kind the class-anchored rule is for, and Sec. 5.5 shows it working. Prompts move the two capabilities in opposite directions. On ECUSTFD, asking for metric extents yields the strongest ordering in the panel ($r = 0.17$ per row) but the worst absolute error among the five prompt forms. On the mass datasets, chain-of-thought costs both.

The two families of results are reconciled by one number. The median within-class spread of the ground truth is $\times 1.08$ on MRGBD, $\times 1.12$ on ECUSTFD, $\times 1.40$ on OmniFood8K and $\times 1.68$ on Nutrition5k: two apples in ECUSTFD differ by about ten percent, two plates of rice in Nutrition5k by nearly a factor of two. OmniFood8K spreads less than Nutrition5k yet scores higher, which argues against a simple dose. Inside each mass dataset, the evidence is mixed (Table 47): across terciles of class spread, the panel's median within-class correlation is flat on Nutrition5k (0.60, 0.68, 0.61) and rises on OmniFood8K (0.62, 0.75, 0.83). We therefore place a change between $\times 1.1$ and $\times 1.4$ without claiming its shape above it. The two groups also differ in target, scene, prompt and class definition, so the design bounds where the change happens rather than isolating its cause. The claim the data support is narrow: on benchmarks where the category leaves little to explain (ECUSTFD, MRGBD, NOCS), no VLM at any scale beats the class prior in log error, and on ECUSTFD and MRGBD none adds within-class information either; on benchmarks that do not, most rank the instance and several beat it, and what they lack is calibration to metric units.

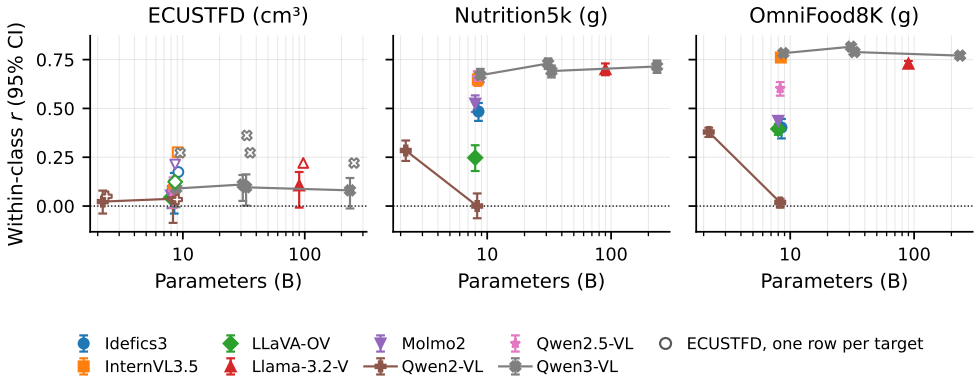

Figure 9: **Model size does not move the within-class correlation; the benchmark does.** Within-class correlation against model size, one point per member (4-bit reruns omitted), with target-clustered 95% intervals, scored per row; on ECUSTFD the open markers aggregate each target to one row. Near zero at every scale on ECUSTFD and near 0.7 on the two mass datasets.

G.2 Did the panel memorise the benchmarks?

All the datasets are public, so a panel member may have seen them in training. A member that had memorised a dataset's ground truth would return it: its answers would pile up at the true value, beyond what its own error distribution predicts. Skill alone moves mass towards the truth smoothly

Table 40: **What the vision–language channel can and cannot resolve, across the panel.** Mean ARE, within-class correlation scored per row (target-clustered 95% interval, **bold** excluding zero) and ΔR^2 beyond the category (**bold** where its target-clustered interval lies above zero; a fourth digit where it rounds to zero). Mass targets (grams) are never pooled with the volume datasets; MRGBD's nine targets allow only point estimates; the 235B member runs in its released FP8.

Model	ECUSTFD (cm^3)			MRGBD (cm^3)			Nutrition5k (g)			OmniFood8K (g)		
	ARE↓	r_{wc} ↑	gross ΔR^2 ↑	ARE↓	r_{wc} ↑	gross ΔR^2 ↑	ARE↓	r_{wc} ↑	gross ΔR^2 ↑	ARE↓	r_{wc} ↑	gross ΔR^2 ↑
Qwen2-VL-2B	3.112	0.02 [-0.04,0.08]	+0.0003	1.398	-0.11	+0.010	0.805	**0.28 [0.23,0.34]**	**+0.040**	0.591	**0.38 [0.35,0.40]**	**+0.072**
Qwen2-VL-7B	2.517	0.04 [-0.09,0.13]	-0.002	0.491	0.16	-0.003	0.716	0.00 [-0.06,0.06]	-0.002	0.593	0.02 [-0.01,0.04]	+0.0001
Qwen2.5-VL-7B	0.504	0.07 [-0.01,0.15]	-0.002	0.477	-0.03	-0.005	0.604	**0.66 [0.62,0.69]**	**+0.201**	0.338	**0.60 [0.57,0.64]**	**+0.163**
LLaVA-OneVision-7B	0.724	0.04 [-0.01,0.09]	+0.0001	0.312	0.00	-0.002	0.782	**0.25 [0.18,0.31]**	**+0.027**	0.526	**0.40 [0.36,0.42]**	**+0.079**
Qwen3-VL-8B	0.581	0.09 [-0.01,0.15]	-0.002	0.323	0.08	+0.017	0.815	**0.67 [0.63,0.70]**	**+0.216**	0.388	**0.78 [0.77,0.79]**	**+0.284**
Qwen3-VL-8B (NF4)	0.567	0.05 [-0.00,0.10]	-0.0004	0.384	-0.09	-0.011	0.948	**0.66 [0.62,0.70]**	**+0.215**	0.343	**0.79 [0.78,0.80]**	**+0.290**
Molmo2-8B	0.819	0.05 [-0.01,0.10]	+0.0005	0.572	0.23	+0.024	0.797	**0.52 [0.48,0.57]**	**+0.134**	0.360	**0.44 [0.41,0.46]**	**+0.094**
InternVL3.5-8B	1.925	**0.08 [0.01,0.13]**	+0.002	0.439	0.05	-0.001	1.009	**0.65 [0.62,0.68]**	**+0.202**	0.389	**0.76 [0.74,0.78]**	**+0.269**
Idefics3-8B	0.961	0.08 [-0.04,0.17]	+0.002	0.748	0.03	-0.010	1.014	**0.48 [0.44,0.53]**	**+0.116**	0.395	**0.40 [0.35,0.45]**	**+0.080**
Qwen3-VL-30B-A3B	0.532	**0.11 [0.03,0.16]**	+0.001	0.227	0.20	+0.066	0.727	**0.73 [0.70,0.76]**	**+0.261**	0.464	**0.82 [0.80,0.83]**	**+0.305**
Qwen3-VL-32B	0.519	**0.10 [0.00,0.16]**	+0.0001	0.358	-0.03	-0.030	0.473	**0.69 [0.66,0.72]**	**+0.236**	0.264	**0.79 [0.77,0.80]**	**+0.290**
Qwen3-VL-32B (NF4)	0.557	**0.10 [0.01,0.17]**	+0.0001	0.332	0.01	-0.037	0.501	**0.68 [0.65,0.71]**	**+0.226**	0.255	**0.79 [0.77,0.80]**	**+0.291**
Llama-3.2-90B-V	0.790	0.11 [-0.01,0.17]	-0.005	0.336	0.24	+0.096	0.442	**0.70 [0.67,0.73]**	**+0.240**	0.344	**0.73 [0.71,0.74]**	**+0.250**
Qwen3-VL-235B-A22B	0.718	0.08 [-0.01,0.14]	+0.0002	0.237	0.08	+0.001	0.786	**0.72 [0.68,0.75]**	**+0.249**	0.561	**0.77 [0.75,0.78]**	**+0.274**

and makes no spike at zero error. From the cached answers on the five datasets whose classes hold several targets (the panel's per-item answers on MADIMA23, MetaFood3D and YCB enter only pools) we therefore computed, per member and dataset, the share of answers within 1% of the truth, the same share after exchanging the member's answers between items of one class, and the density of relative errors in $[0, 1\%]$ over that in $(1\%, 5\%]$ (Table 41). The statistics were fixed before the first run; the analysis is exploratory, and a null result is absence of evidence for verbatim recall, not proof that a model never saw the data.

On ECUSTFD, NOCS, Nutrition5k and OmniFood8K between 0.80% and 1.53% of answers land within 1% of the truth, against 0.87% to 1.45% for the shuffled answers, and the spike ratio is 0.78 to 0.99: there is no excess at the true value. One member each on ECUSTFD, NOCS and Nutrition5k, and eight on OmniFood8K, exceed their shuffled rate at $p < 0.05$; on the mass datasets a within-class correlation of about 0.7 implies they should, and the error density stays smooth. MRGBD is the instructive exception. Its spike ratio is 20, but its shuffled rate (8.05%) matches its hit rate (8.46%) and no member exceeds chance: MRGBD's ground truths are round numbers and the models answer in round numbers, so a member's vocabulary hits them whichever item it is shown. The probe finds no evidence that the panel's results rest on recalled ground truth.

Table 41: **A memorisation probe of the panel's cached answers.** *Hit*: the share of answers within 1% of the ground truth, pooled over members; *Shuffled*: the same after exchanging each member's answers between items of one class ($1{,}000$ draws), the rate its answer vocabulary yields by chance; *Spike*: the density of relative errors in $[0, 1\%]$ over that in $(1\%, 5\%]$, about 1 for a smooth error distribution and far above it for recalled values; the largest member-level hit rate; and the members whose hit rate exceeds their shuffled rate at $p < 0.05$.

Dataset	Members	Items	Hit	Shuffled	Spike	Largest member hit	Members $p < 0.05$
ECUSTFD	13	1,379	1.34%	1.45%	0.86	3.8%	1
MRGBD	13	191	8.46%	8.05%	20.00	38.2%	0
Nutrition5k	14	1,347	1.16%	0.87%	0.78	2.2%	1
NOCS	12	1,500	0.80%	1.43%	0.99	5.3%	1
OmniFood8K	14	8,036	1.53%	1.23%	0.98	2.8%	8

G.3 The VLM channel: geometry or recognition?

If the semantic channel dominates, the natural objection is that a modern VLM would perform better, since it both recognises and reasons. We test this directly using a local Qwen2.5-VL-7B-Instruct, prompted to name the object and provide its volume in cubic centimetres, across $8{,}074$ predictions for seven dataset/condition pairs (Table 43).

On the volume benchmarks, the model never beats the category prior where the prior is estimable (the panel of Sec. 5.3 qualifies this for the two mass datasets). Against the geometric chain, the comparison is one of failure modes rather than of ranks: the chain's median error is below the model's on every dataset the two share, but its mean is dragged past the model's on ECUSTFD by

Table 42: **Fusing the panel with a fixed rule does not help.** The panel combined by the median, against the best member, the per-item oracle and the category prior, all on the items every member answered. *Median−best*: the fused excess; *disagr.*: the members' mean log-spread. The oracle is a bound, not a competitor. The prior here is fitted within the items every member answered, which on YCB is 53 of 800, so it differs from Table 18.

		Mean ARE↓					
Dataset	M	**best**	**median**	**oracle**	**prior**	**median−best**	**disagr.**
ECUSTFD	14	0.466	0.474	0.161	0.156	+0.007	0.79
MADIMA23	13	0.546	2.123	0.199	0.529	+1.577	1.09
MRGBD	14	0.227	0.242	0.045	0.117	+0.015	0.70
MetaFood3D	13	2.519	9.038	1.151	1.772	+6.519	1.20
YCB	9	3.443	5.636	0.482	8.705	+2.193	0.95
NOCS	13	0.473	0.544	0.169	0.580	+0.070	0.79
Nutrition5k	14	0.441	0.589	0.163	0.554	+0.148	0.57
OmniFood8K	14	0.259	0.257	0.082	0.292	-0.002	0.68

the single degenerate item of Sec. 5.2 (6.73 against 0.48 on their 803 shared items, medians 0.35 against 0.39), while on MADIMA23 and MRGBD the chain leads on both statistics. More revealing than the error is its structure. Fig. 10 plots prediction against truth: the output collapses onto a handful of round constants. On ECUSTFD, the model emits 18 distinct values across 2,867 items, with 150 cm^3 accounting for 70% of them; on MRGBD, 5 values across 191 items, 84% of them 150.

The decisive measure is the within-class correlation between prediction and truth, the part of the output that the category cannot explain. We compute it on class-centred log-volumes and bracket it with a bootstrap that resamples whole target instances, since repeated views of one object are not independent observations. On ECUSTFD, $r = 0.073$ (95% interval $[-0.009, 0.147]$ over the 143 distinct targets in the scored crops; the pooled estimate over all items, reported in Table 43, is 0.05); on MRGBD, $r = -0.031$, where nine targets are too few for an interval. The ECUSTFD interval includes zero, and even its upper end would account for $r^2 = 2\%$ of the within-class variance: the model's instance-level information is negligible.

Supplying the ground-truth category settles the matter, because it removes recognition as a confound by construction rather than by argument. Doing so reduces the model's absolute error, sharply on MADIMA23 (0.948 to 0.556) and MetaFood3D (4.858 to 2.678), marginally on MRGBD (0.477 to 0.462), while leaving the within-class correlation unchanged: $r = 0.002$ on MRGBD, compared with -0.031 when the model must recognise the object itself. Knowing the category makes the category-to-volume lookup more accurate and does not help distinguish one apple from another; recognition is not the bottleneck on this benchmark.

We also tested whether an observable metric reference helps by widening the ECUSTFD crop to include the dataset's 1-yuan coin, which has a known diameter. It does not improve accuracy: the mean ARE moves in the wrong direction, from 0.504 to 0.513. It does, however, have a measurable effect, and it is the only intervention we tested that adds semantic source-instance information it did not previously carry. The coin shifts the modal output from 150 to 100 cm^3, cuts the outlier rate from 8.6% to 1.5%, and raises the within-class correlation from $r = 0.073$ $[-0.009, 0.147]$ to $r = 0.112$ $[0.015, 0.184]$ over the same 143 targets, which is the difference between an interval that includes zero and one that does not. The effect is small: r^2 rises from 0.5% to 1.3% of the within-class variance, and the ordering does not translate into accuracy. A visible reference therefore moves the model from no measurable instance information to barely measurable instance information. The negative conclusion of this section rests on the magnitude of that statistic rather than on its sign.

We add the VLM to the benchmark as a first-class channel rather than a separate baseline, creating a heterogeneous geometry+VLM pool alongside the homogeneous backbone, operator, view, sensor and support axes. Paired with the geometric chain, it is a useful fusion source: its errors are essentially uncorrelated with the chain's (-0.02 on ECUSTFD, 0.00 on MRGBD, -0.09 on MADIMA23). The two-source oracle reaches 0.246 against a best single member of 0.483 on ECUSTFD, and the rule chosen on the training folds improves on the best single member by 26% on ECUSTFD (0.483 to 0.359), 25% on MADIMA23 (0.493 to 0.371) and 5% on MRGBD (0.350 to 0.333) on the items each pool shares. Which member the rule improves differs by dataset – on ECUSTFD the best single member is the model, the chain's mean held hostage by one degenerate item, and a gate that routes around that item is exactly what the learned rule provides. Those item

sets are stated because they are not the whole dataset: 803 of ECUSTFD's 837 (App. D.1) and the 40 MADIMA23 images the model scored, on which the chain reads 0.493 rather than the 0.611 it reaches over all 82. The control in Sec. 5.4 shows what that gain is. Replacing the VLM with a *constant* equal to its modal answer, 150 cm^3, yields 0.328, 0.337 and 0.294 on the same three pools, and the cross-validated global median yields 0.349, 0.304 and 0.270: on every dataset, the constant pairing is at least as accurate as the model pairing. The member-level residue is tested in Sec. 5.4: paired against its own modal constant, the model is better on ECUSTFD by a small margin (-0.087 $[-0.163, -0.026]$) and worse on MADIMA23, and the chain already carries the same instance information, so pairing with the model buys nothing that a constant does not. A source that returns 150 on 55–84% of items and draws on only 3 to 18 distinct answers across a whole dataset is nearly a constant; its low error correlation with anything is a property of constants, and averaging geometry with it shrinks the chain's gross over-estimates towards a plausible scale, the classical trade-off of bias against variance (James & Stein, 1961). The gain is real and untrained, but it is shrinkage, not information: to the precision these benchmarks support, the VLM contributes nothing that a single well-chosen number does not. Where the semantic source does carry instance-level information, Sec. G.6 runs the geometric chain on the two mass datasets and finds that, once calibrated per class, it adds one point to the semantic pair.

Table 43: **The VLM channel: geometry or recognition?** A local Qwen2.5-VL-7B asked to name and size the object. *Recog.*: correct-name rate; *within-class* r: correlation with the truth among same-class items; *disp. ratio*: predicted against true within-class spread; the last two columns are the error correlation with the chain and the oracle of a VLM+geometry pool. Columns left of the VLM ARE are over the n crops scored, those right of it over the n_{shared} items the channels share. The prior here is fitted within the scored subset, so it can differ from Table 49's on a subset with few targets.

Dataset	Crop	n	n_{shared}	VLM ARE↓	Geom. ARE↓	Sem. prior↓	Recog.↑	Within-class r ↑	Disp. ratio	Err. corr. with geom.	Oracle VLM+geom↓
ECUSTFD	object	2867	803	0.504	6.728	0.148	63.9%	0.05	1.90	-0.02	0.246
ECUSTFD	reference	2812	789	0.513	6.839	0.151	66.2%	0.13	3.89	-0.02	0.273
MADIMA23	full	40	40	0.948	0.493	0.442	25.0%	–	0.00	-0.09	0.408
MRGBD	full	191	191	0.477	0.350	0.117	100.0%	0.06	3.07	-0.00	0.256
MetaFood3D	full	34	34	4.858	1.597	1.673	32.4%	–	–	0.27	0.931
NOCS	object	1477	1477	0.681	2.856	0.538	86.4%	0.24	2.48	-0.01	0.414
YCB	object	653	653	3.404	9.273	2.152	50.5%	-0.00	–	0.00	0.938

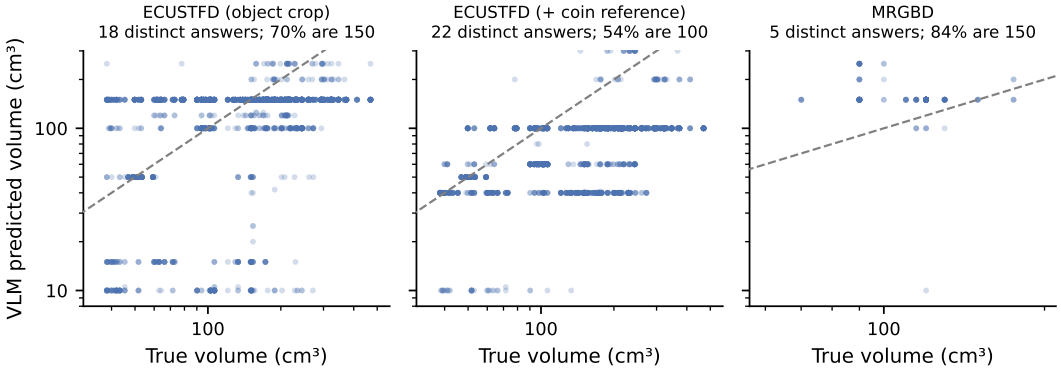

Figure 10: **The incumbent model's answers collapse onto a few round constants.** Qwen2.5-VL-7B's predictions versus ground truth (log-log; the dashed line is the identity). The output collapses onto a few round constants, independent of the true volume, and adding a coin of known size to the crop shifts the constants without creating a trend.

G.4 Prompt forms for the vision–language panel

Table 44 lists each prompt form and its result. A dash marks a form a dataset cannot pose: metric extents yield a volume, and neither mass target assumes a density; OmniFood8K's classes are opaque dish identifiers, so none can be supplied; and the metric reference, a coin in frame, exists only in ECUSTFD. All members are queried using greedy decoding on identical frozen crops and a fixed JSON-only schema, so the comparison is between prompts, not decoding settings. Supplying the ground-truth category reduces absolute error on the three datasets in which recognition is

Table 44: **On the volume benchmarks, no prompt formulation converts the semantic channel into a metric one.** Per dataset: mean ARE↓ (first line) and within-class correlation↑ (second line, instance-clustered 95% interval, **bold** excluding zero). "–": a form the dataset cannot pose (Sec. G.4); §: point estimate without an admissible interval.

Dataset	direct	metric extents	chain-of-thought	class supplied	+ metric reference
ECUSTFD	0.504	0.888	0.670	0.564	0.513
r_{wc}	0.07 [-0.01,0.15]	**0.17 [0.06,0.22]**	**0.08 [0.00,0.13]**	**0.08 [0.01,0.13]**	**0.11 [0.01,0.18]**
MADIMA23	0.948	4.867	2.043	0.556	–
r_{wc}	0.12§	-0.09§	0.41§	0.24§	–
MRGBD	0.477	0.774	0.410	0.462	–
r_{wc}	-0.03§	0.29§	0.03§	0.00§	–
MetaFood3D	4.858	20.579	2.424	2.678	–
r_{wc}	–	–	–	–	–
YCB	3.404	–	–	–	–
r_{wc}	0.00§	–	–	–	–
NOCS	0.681	1.234	0.968	0.967	–
r_{wc}	0.44§	0.36§	0.39§	0.47§	–
Nutrition5k	0.604	–	1.027	0.655	–
r_{wc}	**0.66 [0.62,0.69]**	–	**0.36 [0.31,0.41]**	**0.62 [0.58,0.66]**	–
OmniFood8K	0.338	–	0.490	–	–
r_{wc}	**0.60 [0.57,0.64]**	–	**0.36 [0.33,0.39]**	–	–

hard (from 0.948 to 0.556 on MADIMA23, from 4.858 to 2.678 on MetaFood3D, from 0.477 to 0.462 on MRGBD) and worsens it on ECUSTFD, NOCS and Nutrition5k, where it is not; what it leaves unchanged everywhere is the within-class correlation, which is the measurement that removes recognition as a confound by construction rather than by argument.

ECUSTFD is the only volume benchmark that provides sufficient targets and multi-target classes for the correlation to span an interval, so it is the only column in this table where a prompt effect can be read, and it does not support a flat null. The five intervals are nominal and uncorrected, so this paragraph is exploratory in the sense of Sec. C.6. Three of the five conditions there exclude zero and a fourth touches it: chain-of-thought at 0.08 $[0.00, 0.13]$, class supplied at 0.08 $[0.01, 0.13]$, the added metric reference at 0.11 $[0.01, 0.18]$, and metric extents at 0.17 $[0.06, 0.22]$, against the plain direct prompt's 0.07 $[-0.01, 0.15]$, which does not. Asking for metric extents therefore produces the strongest instance ordering in the whole panel and simultaneously the worst absolute error in it (0.888 against 0.504), which is the cleanest statement we can make of the paper's central asymmetry: what orders instances and what estimates their size are not the same capability, and a prompt can buy the first by spending the second. Prompts do move the correlation. But every prompt that moves it leaves it beneath $r^2 = 3\%$ of the within-class variance, and none converts the movement into a usable volume.

The two mass datasets, where the correlation does carry an interval, introduce a prompt effect of their own, but in the opposite direction. Chain-of-thought costs both accuracy and ordering on both datasets: on Nutrition5k the ARE worsens from 0.604 to 1.027 and the correlation falls from 0.66 $[0.62, 0.69]$ to 0.36 $[0.31, 0.41]$; on OmniFood8K the ARE worsens from 0.338 to 0.490 and the correlation falls from 0.60 $[0.57, 0.64]$ to 0.36 $[0.33, 0.39]$, with the two intervals disjoint on both datasets. Where a channel finally carries signal, asking the model to reason its way to the number loses part of it.

Table 45 gives the strings verbatim, and Table 46 gives the repository identifier of every member.

G.5 Is the saturation contrast an artefact of view multiplicity?

Sec. 5.3 reads a contrast across datasets: the panel ranks portions well where a class varies internally and only weakly where it does not. We must rule out the paper's own subject as a confound. Every benchmark on the saturated side carries many rows per physical target in the panel's crops – ECUSTFD 20, MRGBD 21, NOCS 100 – while both benchmarks on the resolving side carry exactly one by construction. Ground truth is constant within a target, and the model's answer is not, so the within-target variance of the prediction enters the residual variance and contributes no covariance: classical attenuation, which alone predicts the direction of the whole contrast.

We measure the confound rather than argue about it. The top half of Table 47 gives the attenuation factor $\sqrt{B/(B+W)}$ from the between- and within-target sums of squares of the log prediction,

Table 45: **The exact prompt strings.** A query is the context sentence for the crop, a line break, and the form's question, sent with the image under greedy decoding; the new-token caps are 48 for the direct and class-supplied forms, 80 for extents, and 320 for chain-of-thought. `{klass}` is replaced by the item's category string; line breaks inside a prompt are typeset as spaces here, and the strings in the released code are authoritative. The mass datasets use the overhead-plate context and the grams schema; the extents form is defined only for volume.

Context, by crop	
Full	`This photo shows a scene containing one main object.`
Object	`This is a cropped photo of a single object.`
Reference	`This photo shows a single object next to a coin of known size (a Chinese 1-yuan coin, 25 mm across). Use the coin to establish scale.`
Mass datasets	`This is an overhead photo of a plate of food.`
Question, volume	
Direct	`Estimate the physical volume of the object in cubic centimetres. Reply with ONLY JSON: {"object": "<name>", "volume_cm3": <number>}`
Metric extents	`Report the object's bounding-box extents in centimetres and the fraction of that box the object actually fills. Reply with ONLY JSON: {"object": "<name>", "length_cm": <number>, "width_cm": <number>, "height_cm": <number>, "fill_factor": <0-1>}`
Chain-of-thought	`Estimate the physical volume of the object in cubic centimetres. First reason step by step: identify the object, recall its typical size, judge how large this particular one is relative to typical, then compute. End your reply with ONLY this JSON on the last line: {"object": "<name>", "volume_cm3": <number>}`
Class supplied	`The object in this photo is a {klass}. Estimate THIS particular instance's physical volume in cubic centimetres -- do not simply report the typical size for a {klass}. Reply with ONLY JSON: {"object": "{klass}", "volume_cm3": <number>}`
Question, mass	
Direct	`Estimate the total mass of the food on this plate, in grams. Reply with ONLY JSON: {"object": "<main food>", "mass_g": <number>}`
Chain-of-thought	`Estimate the total mass of the food on this plate, in grams. First reason step by step: identify the food, judge the portion size relative to a typical serving, then compute. End your reply with ONLY this JSON on the last line: {"object": "<main food>", "mass_g": <number>}`
Class supplied	`The food on this plate is {klass}. Estimate THIS particular serving's total mass in grams -- do not simply report a typical portion of {klass}. Reply with ONLY JSON: {"object": "{klass}", "mass_g": <number>}`

Table 46: **The panel's members.** The report each model was released with, the repository identifier of the weights as run, its vision encoder and its language model; NF4 variants load the same weights in 4-bit NormalFloat, and the 235B member is the FP8 release. Molmo2 is cited through the Molmo family paper and Llama-3.2-Vision through the Llama 3 report. Greedy decoding, one fixed JSON-only prompt per form (Table 45), identical frozen crops.

Member	Ref.	Repository	Vision encoder	Language model
Qwen2.5-VL-7B	(Bai et al., 2025b)	`Qwen/Qwen2.5-VL-7B-Instruct`	Qwen2.5-ViT	Qwen2.5-7B
Qwen3-VL-8B	(Bai et al., 2025a)	`Qwen/Qwen3-VL-8B-Instruct`	Qwen3-ViT	Qwen3-8B
InternVL3.5-8B	(Wang et al., 2025b)	`OpenGVLab/InternVL3_5-8B-HF`	InternViT	Qwen3-8B
LLaVA-OneVision-7B	(Li et al., 2025)	`llava-hf/llava-onevision-qwen2-7b-ov-hf`	SigLIP-SO400M	Qwen2-7B
Idefics3-8B	(Laurençon et al., 2024)	`HuggingFaceM4/Idefics3-8B-Llama3`	SigLIP-SO400M	Llama-3.1-8B
Qwen3-VL-8B (NF4)	(Bai et al., 2025a)	`Qwen/Qwen3-VL-8B-Instruct`	Qwen3-ViT	Qwen3-8B
Qwen3-VL-32B (NF4)	(Bai et al., 2025a)	`Qwen/Qwen3-VL-32B-Instruct`	Qwen3-ViT	Qwen3-32B
Qwen2-VL-2B	(Wang et al., 2024)	`Qwen/Qwen2-VL-2B-Instruct`	Qwen2-ViT	Qwen2-1.5B
Qwen2-VL-7B	(Wang et al., 2024)	`Qwen/Qwen2-VL-7B-Instruct`	Qwen2-ViT	Qwen2-7B
Qwen3-VL-32B	(Bai et al., 2025a)	`Qwen/Qwen3-VL-32B-Instruct`	Qwen3-ViT	Qwen3-32B
Qwen3-VL-30B-A3B	(Bai et al., 2025a)	`Qwen/Qwen3-VL-30B-A3B-Instruct`	Qwen3-ViT	Qwen3-30B-A3B
Qwen3-VL-235B-A22B	(Bai et al., 2025a)	`Qwen/Qwen3-VL-235B-A22B-Instruct-FP8`	Qwen3-ViT	Qwen3-235B-A22B
Llama-3.2-90B-V	(Grattafiori et al., 2024)	`meta-llama/Llama-3.2-90B-Vision-Instruct`	CLIP ViT-H	Llama-3.2-90B
Molmo2-8B	(Deitke et al., 2025)	`allenai/Molmo2-8B`	CLIP ViT-L	OLMo-2-8B

computed *within* class. Where each target contributes several rows, Spearman's correction for attenuation (Spearman, 1904), with the observed target mean as the true score, relates the two units:

$$r_{\text{row}} = r_{\text{target}}\sqrt{\rho}, \qquad \rho = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{between}} + \sigma^2_{\text{within}}}, \tag{6}$$

where ρ is the between-target share of the source's prediction variance within class. It holds exactly when targets contribute equal numbers of rows, and both correlations are taken over one class set. The table also recomputes every within-class correlation with each target aggregated to one row. Over the thirteen ECUSTFD members on their direct prompts, the median member moves from 0.08 per row to 0.22 per target, and four of the thirteen target-unit intervals exclude zero, at $r = 0.27$–0.36; with the three prompt ablations of App. G.4 included, sixteen runs, the median is 0.24 and six intervals exclude zero, up to 0.41 for the metric-extents prompt. Within-class attenuation is 0.29, and pooled across classes it is 0.66. Eq. (6) needs both correlations on one class set, so we recompute them on the classes holding at least three distinct targets: there the median member reads 0.06 per row and 0.18 per target, with an attenuation of 0.28, and the equation reproduces every member's row-unit correlation to within 0.021, the residual coming from targets that contribute unequal numbers of rows. On the two mass datasets, each target contributes one row, so the two units coincide there and nothing is attenuated. The claim in Sec. 5.3 is therefore stated at the target unit, not the row unit.

We also asked the question within a single dataset, where the scene and prompt are held fixed, and only the spread varies. Binning ECUSTFD's classes by their own within-class coefficient of variation, the median member's correlation is 0.13 in the tight half and 0.19 in the loose half, and neither median member's own interval excludes zero. The loose half is higher for 7 of the 13 members (median difference $+0.01$, one-sided sign test $p = 0.50$; with the prompt ablations, 10 of 16 runs and $p = 0.23$), descriptive only since the members share images and prompts. A dose-response within the dataset is therefore not supported; reading one from a single model, or from one of several intervals excluding zero, would be the overlapping-intervals argument this paper rejects elsewhere.

Nor would it have identified what we claimed. The bins are cut on the same ground-truth spread that enters the correlation, so a model of constant per-item error would show the same rise by range restriction alone. That is not a rival explanation to the one in Sec. 5.3; it is the same one. The substitution condition of Sec. 3.2 says a channel can replace the class prior only where σ_w exceeds its own error, and σ_w *is* the range. What the saturated benchmarks establish is therefore a fact about what they permit to be observed, not a fact about what vision–language models can do, and Sec. 5.3 claims only the former.

G.6 The geometric channel on a mass dataset

Nutrition5k is the only dataset in this study that varies within class (Sec. 5.3) and includes a depth sensor, so it is where the question of Sec. 5.5 can be put to geometry with room to answer. We ran the reference chain on the overhead RealSense frame of each of its 1,347 dishes: the sensor arm and the four metric backbones, geom operator, GroundingDINO-plus-SAM support, under the uniform scale protocol of Sec. D.4. Three conventions differ from the volume datasets. The target is mass in grams, so a mean ARE of a volume in cm^3 assumes a density of $1\ \text{g/cm}^3$ and is labelled as such; the statistics the comparison rests on, within-class correlation and ΔR^2 in log space, fit a slope and are unit-free. Folds and intervals are grouped on the dish, as the panel's rows are (App. J.4). Nutrition5k dishes were scanned incrementally, so consecutive ids can be one plate photographed as an ingredient was added; regrouping on the plate session, dishes at most five minutes apart with the same dominant ingredient (1,133 sessions over the 1,347 dishes), moves no statistic below by more than the width of its interval (sensor arm $r = 0.41$, $\Delta R^2 = +0.080$; best VLM 0.70, $+0.245$; category prior ARE 0.533 against 0.552). And the camera intrinsics are not distributed, so we use those of the RealSense D400-series colour stream at 640×480 ($54.7^\circ \times 42.5^\circ$; the depth map is aligned to the colour frame, as the in-mask depth statistics show, with object pixels closer than the plate and no misalignment band at the silhouette), which scales every volume by one common factor and leaves the unit-free statistics untouched. Two checks bound the assumption: the sensor's in-mask depth has a median of 0.34 m and its plate-level median lies between 0.35 and 0.41 m across dishes, consistent with a rig the dataset documents as under 0.4 m high, and the intrinsics enter a volume only through the field of view, so a 5% error in the focal length moves every volume by 10%

Table 47: **The contrast is not an artefact of view multiplicity, and it survives inside one dataset.** *Top*: the panel's within-class correlation scored per row, as Table 40 reports it, and again with each target aggregated to one row, so every dataset is scored at the unit the two mass benchmarks already use; *Atten.* is the attenuation the row unit imposes, $\sqrt{B/(B+W)}$ over the between- and within-target sums of squares. *Bottom*: the same statistic inside one dataset, with classes binned by their own ground-truth spread, so scene and prompt are held fixed and only the spread varies. Each row is the median over the panel members scored on that dataset, with the members' range in brackets and their count in *Runs*; the class-spread edges are the members' medians. **Bold**: the median member's own interval excludes zero. Both halves read the panel in `runs/vlm_panel/`; *Atten.* is computed within class, since the statistic it corrects is a within-class one, and the pooled version overstates it about twofold on the volume benchmarks.

Dataset	Rows per target	Atten.	r_{wc} per row	r_{wc} per target	Runs
ECUSTFD	20.0	0.29	0.08	0.24	16
MADIMA23	1.2	0.61	0.22	0.31	10
MRGBD	21.2	0.27	0.04	–	16
YCB	18.9	0.00	0.00	–	9
NOCS	99.9	0.51	0.36	–	15
Nutrition5k	1.0	1.00	0.64	0.64	21
OmniFood8K	1.0	1.00	0.73	0.73	15

Within one dataset, classes binned by their own spread

Dataset	Class CV	n	Classes	Runs	r_{wc}
ECUSTFD	tightest (0.03–0.12)	74	9	16	0.13 [-0.08, 0.31]
ECUSTFD	loosest (0.12–0.54)	67	9	16	0.24 [-0.03, 0.46]
Nutrition5k	tightest (0.18–0.44)	364	16	21	**0.60** [0.01, 0.72]
Nutrition5k	middle (0.44–0.56)	458	16	21	**0.68** [0.04, 0.74]
Nutrition5k	loosest (0.57–1.23)	525	17	21	**0.61** [-0.04, 0.76]
OmniFood8K	tightest (0.08–0.24)	3122	53	15	**0.62** [0.06, 0.78]
OmniFood8K	middle (0.24–0.33)	3810	53	15	**0.75** [-0.03, 0.81]
OmniFood8K	loosest (0.33–1.40)	974	54	15	**0.83** [0.10, 0.87]

and nothing else in the table. Two depth files are in millimetres, whereas the rest use the 10^{-4} m unit the dataset documents; we converted them to a common unit before the run.

Geometry resolves instances here. The sensor arm reaches a within-class correlation of 0.41 $[0.36, 0.47]$ and explains $\Delta R^2 = +0.043$ $[+0.020, +0.067]$ of the log-mass variance beyond the category, net of the category prior's own null; the four backbones lie between $+0.027$ and $+0.051$ net, every interval excluding zero (Table 48). This is the only dataset in the study where the geometric channel separates from that null at all: on ECUSTFD, the volume benchmark with the most room, the same channel's net contribution is $+0.002$ $[-0.056, +0.059]$. That is the point Sec. 5.5 makes, in its sharpest form: the volume benchmarks leave the channel no room to show what it carries, and where a benchmark does leave room, the channel shows it. What geometry does not do is calibrate. Its median bias runs from $\times 0.38$ to $\times 1.95$ across backbones and its mean ARE from 1.2 to 3.7 (5.3 for the sensor arm), against 0.44 for the best VLM member and 0.55 for the category prior, so on the absolute scale it is the weakest source on the table; and the VLM's within-class correlation of 0.70 and net ΔR^2 of $+0.205$ leave it a distant second on the unit-free statistics too.

At the decision level, raw geometry cannot be pooled with the semantic sources at all: it emits cm^3, and they emit grams, so a mean or a weighted mean of the two is not a quantity, and we report no such pool. The two homogeneous pools behave as on the volume datasets: the four backbones fuse to 1.202 against a best of 1.225, and the sensor arm paired with ZoeDepth to 2.71 against 3.72, a large relative gain between two poor sources. The semantic pair, the VLM with the category prior, reaches 0.357 against the VLM's 0.442, capturing 54% $[44, 62]$ of its headroom, and is the best pool on the table. The question the run can answer is whether geometry, once brought into grams, adds to that pair.

The reading is the one the paper has argued from the other side. Where the benchmark lets a channel be measured, the geometric chain does carry instance information; what it lacks on a mass target is a scale, part metric and part density, and the untrained rules and cue-based selectors of Sec. 3.1 do not recover it from a source whose gross errors dwarf its signal. The per-class calibration of Sec. 3.1, a factor per class learned on the training folds, does that: it maps the geometric estimate onto the mass scale and repairs the centre but not the tail. The calibrated sensor arm's median ARE falls from 0.80 to 0.36, level with the category prior's 0.35, and its median bias to $\times 1.00$, while its mean stays at 3.3 because the gross over-estimates are multiplicative failures that no single factor per class

removes; its within-class correlation (0.39) and ΔR^2 (+0.066) are unchanged within their intervals, as a per-class rescale must leave them. Calibrated geometry with the VLM and the prior reaches 0.346 against the pair's 0.357 (one percentage point), at the edge of the tolerance this paper uses for an improvement, while calibrated geometry with the prior alone (0.563) does not beat the prior. Calibrating the VLM the same way, since it is the source that ranks but does not calibrate, lowers its own error from 0.442 to 0.427 (median 0.39 to 0.31, bias to $\times$1.00) but leaves the pair at 0.355 and the triple at 0.346: the per-class factor helps the source alone and not the pool. The best member is chosen on the evaluation set; with the incumbent Qwen2.5-VL-7B fixed in advance, whose own error is 0.604, the pair reaches 0.438 and calibrated geometry brings it to 0.400, four points, so geometry adds more to a weaker semantic member. The errors of the three sources are only weakly correlated (mean pairwise 0.15), and the per-item oracle over the calibrated triple falls to 0.190, so the information is in the pool; what the untrained rules and the cue-based selector recover of it is one point.

Table 48: **The geometric channel on Nutrition5k.** One overhead RGB-D frame per dish, 1,347 dishes, target mass in grams. *Top*: each source alone. ARE treats the volume as a mass at 1 g/cm^3; the median bias is the median ratio of prediction to truth; within-class correlation and ΔR^2 are fitted in log space and do not depend on the unit (dish-clustered 95% intervals). ΔR^2 is net of the category prior's null, as in Table 67, which the two prior rows carry themselves. *Bottom*: pools whose members share a unit, scored as in Table 49, with headroom measured against the best single *member*, so a pool containing a smaller one is credited with the total rather than the increment (Sec. G.6).

Source	n	ARE ρ=1↓	Median ARE↓	Median bias	Within-class corr.↑	ΔR^2 net of null↑
UniDepthV2	1,347	1.225	0.762	×0.38	0.37 [+0.32, +0.42]	+0.027 [+0.01, +0.05]
MoGe-2	1,347	2.465	0.879	×1.53	0.42 [+0.37, +0.47]	+0.046 [+0.02, +0.07]
Depth Pro	1,347	3.203	0.930	×1.74	0.43 [+0.38, +0.48]	+0.049 [+0.03, +0.07]
Sensor depth, per-class calibrated	1,347	3.343	0.363	×1.00	0.39 [+0.34, +0.45]	+0.030 [+0.01, +0.05]
ZoeDepth	1,347	3.716	0.979	×1.95	0.43 [+0.38, +0.48]	+0.051 [+0.03, +0.08]
Sensor depth (RealSense)	1,347	5.306	0.803	×1.39	0.41 [+0.36, +0.47]	+0.043 [+0.02, +0.07]
Best VLM member (Llama-3.2-90B-V, per-class calibrated)	1,347	0.427	0.306	×1.00	0.69 [+0.66, +0.72]	+0.194 [+0.16, +0.22]
Best VLM member (Llama-3.2-90B-V)	1,347	0.442	0.389	×0.72	0.70 [+0.67, +0.73]	+0.205 [+0.17, +0.23]
Category prior	1,347	0.552	0.347	×0.99	-0.28 [-0.33, -0.24]	+0.040 (null)
Global prior	1,347	0.864	0.493	×1.03	-0.11 [-0.16, -0.05]	+0.005 (null)

Pool	Src.	Best single↓	Selected rule	Fused ARE↓	Oracle ceiling↓	Headroom captured↑
Depth backbone	4	1.225	Selector (classify)	**1.202**	1.010	11 (-4, 22)
Depth sensor	2	3.716	Selector (regress)	**2.710**	2.537	85 (80, 90)
VLM + prior	2	0.442	Geometric mean	**0.357**	0.283	54 (44, 62)
Calibrated geometry + VLM + prior	3	0.442	Conflict-aware	**0.346**	0.196	39 (29, 47)
Calibrated geometry + prior	2	0.552	Inverse-variance	0.563	0.286	-4 (-26, 10)
Calibrated VLM + prior	2	0.427	Geometric mean	**0.355**	0.277	48 (37, 58)
Calibrated geometry + calibrated VLM + prior	3	0.427	Conflict-aware	**0.346**	0.190	34 (24, 43)

H DECISION-LEVEL FUSION: CEILINGS, RULES AND CONTROLS

This appendix reports every pool's ceiling and what the rules reach, the control sources, and the ablations: dynamic selection, diversity, pool composition and size, the selector's cues, the within-class shuffle, fusion before integration, the pool-gain model and injected conflict.

H.1 FUSION CEILINGS AND WHAT RULES REACH, IN FULL

Table 49 and Fig. 11 report, for each of the 46 pools, the best single source, the rule chosen by nested selection, two trained combiners, and the per-item oracle. The oracle could remove 17–63% of the best source's error on the backbone pools, 37–76% when all channels are pooled, and 73–84% on the two support pools. Two integration operators applied to one depth field leave little for a selector: their errors correlate at 0.56–1.00, and the oracle gain is under 10% on three datasets.

The rules chosen by nested selection recover little. Pooling depth backbones improves on the best backbone only on MADIMA23 (+7%) and MRGBD (+18%). On ECUSTFD it fails catastrophically: the dispersion-weighted mean follows the degenerate item into every estimate. Pooling the two operators helps only on MADIMA23 (+19%). Pairing the chain with the VLM removes 26%, 25% and 5% of the best member's error on ECUSTFD, MADIMA23 and MRGBD, and makes MetaFood3D 24% worse. On the MADIMA23 sensor pool, the selected rule is worse than the better sensor (0.762 against 0.607). On ECUSTFD's two viewpoints, whose errors are nearly independent ($r = 0.05$), the rule captures $H = 30\%$ (95% CI 3–49%) of the headroom. A learned selector was

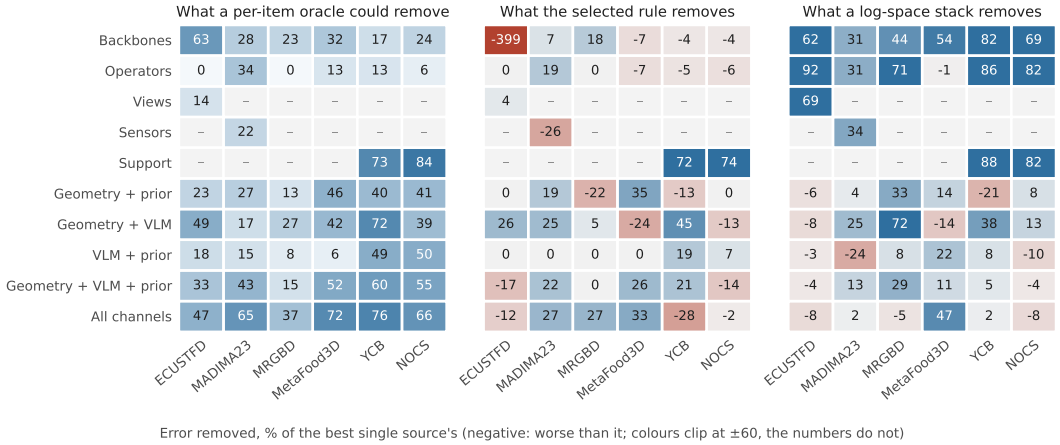

Figure 11: **How much fusion could help, and how much it does.** Error removed, as a percentage of the best single source, per channel axis and dataset: by a per-item oracle, by the rule selected on training folds, and by a least-squares stack fitted in log space. Empty cells are axes a dataset does not offer. A stack can exceed the selection oracle, since a blend is not restricted to one source.

Table 49: **Every reported pool: what the rules reach and what an oracle could.** Per pool: diversity (mean pairwise error correlation), the best single source, the rule selected on training folds with its out-of-fold ARE, two trained combiners scored on the same folds (a least-squares stack in log space and a selector that reads the sources' own outputs), the per-item oracle, and the headroom H the selected rule captures (target-clustered 95% interval). **Bold**: at least 1% below the best single source. *, majority rule over folds that disagree; n/s, oracle gap indistinguishable from zero; §, fails TARGETS; n, the items the members share.

						Fused, out of fold					
Dataset	**Channel axis**	**Src.**	n	**Err. corr.**	**Best single↓**	**Selected rule**	**ARE↓**	**Log stack↓**	**Selector on outputs↓**	**Oracle ceiling↓**	**Headroom captured↑**
ECUSTFD	All channels	6	803	0.41	0.163	Selector (regress)	0.182	0.175	0.182	0.087	-26 (-59, -9)
ECUSTFD	Depth backbone	4	837	0.99	1.417	Inverse-variance*	7.069	**0.537**	6.488	0.528	<-100 (<-100, 25)
ECUSTFD	Geometry + prior	2	837	-0.03	0.159	Inverse-variance*	0.159	0.168	0.166	0.122	-1 (-4, 0)
ECUSTFD	Geometry + VLM	2	803	-0.02	0.483	Selector (classify)*	**0.359**	0.520	**0.330**	0.246	52 (28, 69)
ECUSTFD	Geometry + VLM + prior	3	803	0.08	0.163	Selector (classify)*	0.190	0.168	0.179	0.108	-50 (<-100, -15)
ECUSTFD	Integration operator	2	837	1.00	6.493	Selector (classify)	6.495	**0.536**	6.511	6.474	-8 (-35, 7)
ECUSTFD	Viewpoint	2	834	0.05	1.732	Selector (regress)*	**1.658**	**0.537**	**1.552**	1.486	30 (3, 49)
ECUSTFD	VLM + prior	2	803	0.30	0.163	Selector (classify)	0.163	0.168	0.204	0.133	0 (0, 0)
MADIMA23	All channels	6	39	0.07	0.445	Mean	**0.324**	**0.437**	**0.404**	0.154	42 (0, 64)
MADIMA23	Depth backbone	4	82	0.32	0.611	Selector (regress)	**0.567**	**0.424**	**0.523**	0.440	26 (-12, 52)
MADIMA23	Depth sensor	2	81	0.15	0.607	Inverse-variance*	0.762	**0.401**	0.692	0.473	<-100 (<-100, 28)
MADIMA23	Geometry + prior	2	82	-0.11	0.436	Mean	**0.354**	**0.420**	**0.431**	0.317	69 (-51, >100)
MADIMA23	Geometry + VLM	2	40	-0.09	0.493	Inverse-variance*	**0.371**	**0.370**	0.496	0.408	>100 (41, >100)
MADIMA23	Geometry + VLM + prior	3	40	0.11	0.445	Geometric mean	**0.349**	**0.386**	0.447	0.252	50 (-3, 85)
MADIMA23	Integration operator	2	82	0.63	0.611	Inverse-variance*	**0.496**	**0.421**	**0.519**	0.404	56 (21, 95)
MADIMA23	VLM + prior	2	40	0.64	0.445	Inverse-variance*	0.445	0.553	0.445	0.378	-1 (-3, 0)
MRGBD§	All channels	6	191	0.07	0.117	Mean	**0.085**	0.123	0.128	0.074	74 (13, >100)
MRGBD§	Depth backbone	4	191	0.21	0.162	Mean	**0.133**	**0.090**	0.168	0.125	78 (-32, >100)
MRGBD§	Geometry + prior	2	191	0.09	0.117	Selector (regress)*	0.142	**0.079**	0.129	0.101	<-100 (<-100, 0)
MRGBD§	Geometry + VLM	2	191	-0.00	0.350	Selector (regress)*	**0.333**	**0.098**	**0.274**	0.256	18 (<-100, 61)
MRGBD§	Geometry + VLM + prior	3	191	0.10	0.117	Selector (classify)*	0.117	**0.083**	0.137	0.100	0 (0, 0)
MRGBD§	Integration operator	2	191	0.56	0.350	Selector (classify)	0.350	**0.100**	0.350	0.350	n/s
MRGBD§	VLM + prior	2	191	0.22	0.117	Selector (classify)	0.117	**0.108**	0.117	0.107	n/s
MetaFood3D	All channels	6	34	0.37	1.597	Selector (regress)*	**1.074**	**0.844**	**0.709**	0.451	46 (-39, 86)
MetaFood3D	Depth backbone	4	34	0.56	1.597	Selector (classify)*	1.712	**0.730**	**1.258**	1.084	-22 (-89, 34)
MetaFood3D	Geometry + prior	2	34	0.50	1.597	Selector (regress)*	**1.041**	**1.375**	**0.958**	0.858	75 (-17, 93)
MetaFood3D	Geometry + VLM	2	34	0.27	1.597	Geometric mean*	1.986	1.828	1.671	0.931	-58 (<-100, -3)
MetaFood3D	Geometry + VLM + prior	3	34	0.48	1.597	Selector (regress)*	**1.187**	**1.427**	**0.958**	0.768	49 (<-100, 85)
MetaFood3D	Integration operator	2	34	0.99	1.597	Selector (classify)*	1.707	1.614	1.695	1.395	-54 (<-100, 17)
MetaFood3D	VLM + prior	2	34	0.66	1.673	Selector (classify)	1.673	**1.313**	1.673	1.577	0 (0, 0)
YCB	All channels	6	653	0.40	1.862	Selector (classify)*	2.384	**1.826**	**1.440**	0.449	-37 (-81, 14)
YCB	Depth backbone	4	800	0.98	13.474	Selector (classify)*	14.001	**2.396**	13.449	11.196	-23 (<-100, 43)
YCB	Geometry + prior	2	800	0.06	1.854	Selector (classify)*	2.103	2.243	**1.769**	1.110	-34 (<-100, 21)
YCB	Geometry + VLM	2	653	0.00	3.404	Geometric mean*	**1.873**	**2.098**	**1.393**	0.938	62 (<-100, 85)
YCB	Geometry + VLM + prior	3	653	0.09	1.862	Conflict-aware	**1.469**	**1.768**	**1.483**	0.738	35 (-9, 51)
YCB	Integration operator	2	800	0.90	17.224	Selector (classify)*	18.047	**2.343**	17.534	15.023	-37 (-88, 1)
YCB	Segmentation source	2	800	0.10	17.224	Selector (classify)*	**4.883**	**2.141**	**4.916**	4.621	98 (91, 99)
YCB	VLM + prior	2	653	0.18	1.862	Geometric mean	**1.507**	**1.718**	**1.233**	0.952	39 (5, 69)
NOCS§	All channels	6	1,477	0.34	0.503	Inverse-variance	0.515	0.543	**0.357**	0.172	-4 (<-100, 25)
NOCS§	Depth backbone	4	13,371	0.88	1.922	Selector (regress)*	2.001	**0.592**	**1.644**	1.461	-17 (-42, 15)

Table 49: **(continued)**

Dataset	Channel axis	Src.	n	Err. corr.	Best single↓	Fused, out of fold: Selected rule	ARE↓	Log stack↓	Selector on outputs↓	Oracle ceiling↓	Headroom captured↑
NOCS[§]	Geometry + prior	2	13,371	-0.02	0.521	Inverse-variance	0.521	**0.479**	0.577	0.308	-0 (-4, 1)
NOCS[§]	Geometry + VLM	2	1,477	-0.01	0.681	Inverse-variance*	0.773	**0.590**	**0.575**	0.414	-34 (< -100, 0)
NOCS[§]	Geometry + VLM + prior	3	1,477	-0.04	0.503	Inverse-variance*	0.575	0.521	**0.361**	0.224	-26 (< -100, 28)
NOCS[§]	Integration operator	2	13,371	0.95	3.287	Selector (regress)	3.492	**0.597**	**3.199**	3.079	-98 (< -100, 40)
NOCS[§]	Segmentation source	2	13,371	-0.02	3.287	Selector (regress)*	**0.844**	**0.596**	**0.628**	0.519	88 (19, 96)
NOCS[§]	VLM + prior	2	1,477	-0.09	0.503	Geometric mean*	**0.466**	0.552	**0.359**	0.252	15 (< -100, 84)

Table 50: **Each combiner, summarised over the** 46 **pools.** Change in error against the pool's best single source chosen on the training folds (geometric mean of the ratio, datasets weighted equally; interval from resampling datasets and then pools within them, descriptive with six datasets; **bold** excludes zero), and how many pools the rule improves or worsens by more than 1%. Below the rule: two controls, the best source times one training-fold factor and the log-space constant, and the log stack measured against that constant instead of the best source.

Combiner	Error change	95% interval	Better	Worse
Linear stack (log)	**−46%**	[−61%, −28%]	34	12
Precision-weighted mean	**−26%**	[−45%, −5%]	27	18
Selector on the sources' outputs	**−24%**	[−39%, −8%]	25	13
Nested selection with stack	−22%	[−41%, +12%]	25	14
Nested-selected rule	**−19%**	[−33%, −6%]	20	17
Linear stack (cm^3)	−10%	[−33%, +23%]	24	21
OLA	−4%	[−19%, +10%]	15	22
KNORA-E	+2%	[−8%, +14%]	18	23
KNORA-U	+21%	[−2%, +61%]	16	27
Log stack, against the constant	**−28%**	[−43%, −13%]	40	3
log-space constant	−24%	[−49%, +6%]	21	22
best source, one factor	+8%	[−13%, +32%]	9	37

the majority choice on 26 pools and improved 9 of them; an untrained rule was chosen on 20 and improved 11. The inner folds prefer the selector more often than the outer folds reward it.

The support axis is the exception in size: fusing two segmentation sources takes YCB from 17.22 to 4.88 and NOCS from 3.29 to 0.84, 98% and 88% of the headroom. It is a repair, not a gain over a trivial source. A constant beats both fused pools (1.06 on YCB, 0.42 on NOCS), and on NOCS the second segmenter returns one salient region per frame covering every object, so what the selector learns is when an object fills enough of the frame for that region to serve (App. D.7).

Table 50 summarises every combiner over the 46 pools, datasets weighted equally, against each pool's best single source chosen on the training folds. One combiner is clearly better: the least-squares stack fitted in log space, where these sources' errors live, which lowers error by 46% (95% interval 28–61%) and improves 34 pools. Its intercept is a fitted constant, and the log-space constant alone gains 24% (−6 to 49%) because, on these benchmarks, the best source is often worse than a constant. Against that constant, the stack still lowers error by 28% (13–43%) and improves 40 of the 46 pools, so most of the gain comes from the sources. One training-fold factor on the best source does not help (+8%). The same stack fitted in cubic centimetres is indistinguishable from the best source (−10%, −33 to +23%); the nested-selected rule the tables report lowers error by 19% (6–33%), and KNORA-U makes it worse by 21% (−2 to +61%). The selector that reads the sources' own outputs sits in the same band as the precision-weighted mean, which says that part of the per-item competence signal lies in the sources' disagreement (App. H.3.1). With six datasets, the intervals are descriptive. Under injected conflict, a mis-scaled duplicate of one member at up to 4× its bias, the mean is captured at once, the median and trimmed mean degrade smoothly, and the learned selectors are the most robust (App. H.11).

Why untrained pooling recovers so little is an identity, not an experiment. For a mean combiner, the ensemble's squared error equals the members' mean error minus their spread about the mean (Krogh & Vedelsby, 1995), in whatever space the mean is taken. Diversity is present: the spread removes a median 41% of the members' mean squared log error. The outcome depends on whether members differ in accuracy as much as they differ from each other. The mean beats the pool's best member on 25 of 58 pools, the 46 reported and 12 controls, exactly those on which the spread exceeds the gap between the mean member and the best. Error correlation and realised gain are related in the

expected direction, but weakly: Pearson $r = -0.36$ ($p = 0.014$), Spearman -0.29 ($p = 0.05$), and $r = -0.25$ ($p = 0.10$) without the ECUSTFD backbone pool. We treat the relation as exploratory (App. H.4).

H.2 Control sources: information or shrinkage?

Table 51: **Control sources: is the fusion gain information or shrinkage?** Mean ARE of the chain alone and paired, under nested selection, with the VLM, the class prior, a constant equal to the VLM's modal answer and the cross-validated global prior, all on the items the chain and the incumbent model share. *Model helps*: the VLM pairing is at least 1% below the pool's better member, the only case in which the control asks anything. *Constant/VLM*: the constant pairing's error over the VLM pairing's, with its one-sided 95% upper bound in brackets; **bold**, the bound is below 1.10. Last two columns: constant pairing minus prior pairing, target-clustered 95% intervals (negative favours the constant). §, fails TARGETS.

		Mean ARE↓						Constant/VLM		Constant − prior	
Dataset	n	Geometry	+ VLM	+ prior	+ constant	+ global prior	Model helps	ARE	Log	ARE	Log
ECUSTFD	803	6.728	0.359	0.181	0.328	0.349	yes	**0.91 (0.98)**	**0.84 (0.92)**	+0.148 [+0.057, +0.223]	+0.156 [+0.089, +0.217]
MADIMA23	40	0.493	0.371	0.306	0.337	0.304	yes	0.91 (1.12)	**0.83 (1.02)**	+0.031 [-0.041, +0.106]	+0.004 [-0.076, +0.085]
MRGBD§	191	0.350	0.333	0.142	0.294	0.270	yes	**0.88 (0.97)**	**0.89 (0.93)**	+0.152 [+0.054, +0.199]	+0.110 [+0.024, +0.142]
MetaFood3D	34	1.597	1.986	1.041	1.373	1.041	no	**0.69 (1.03)**	**0.74 (0.92)**	+0.332 [+0.093, +0.641]	+0.034 [-0.097, +0.169]
YCB	653	9.273	1.873	2.370	1.460	2.370	yes	0.78 (1.10)	1.14 (1.38)	-0.910 [-2.125, -0.038]	+0.207 [-0.064, +0.461]
NOCS§	1,477	2.856	0.773	0.538	0.410	0.475	no	**0.53 (0.80)**	**0.73 (0.96)**	-0.129 [-0.465, +0.135]	+0.085 [-0.164, +0.314]

Pairing the chain with the VLM removes 5–26% of the best member's error on ECUSTFD, MADIMA23 and MRGBD. Table 51 asks whether that is information the model carries or shrinkage of the chain's gross overestimates towards a plausible scale. It pairs the chain, on the items chain and incumbent share, with a constant equal to the model's modal answer ($150\,\mathrm{cm}^3$; 250 on NOCS; 100 on YCB) and with the cross-validated global prior, the fully blind version of the same control. The control asks something only where pairing with the model helps. That is ECUSTFD, MADIMA23, MRGBD, and YCB; on MetaFood3D and NOCS, model pairing is worse than the pool's better member, and a constant match would say nothing.

Where the model helps, the constant pairing's error is 0.78–0.91 of the model pairing's under ARE, and the one-sided 95% bounds fall below 1.10 on ECUSTFD and MRGBD, and reach it on YCB (Table 51). ECUSTFD is the only dataset where the model helps pass TARGETS, and the constant stays within that post hoc 10% margin under both metrics. There, the model's contribution to the chain is one a constant supplies, and under the log error the constant is the better partner (one-sided $p = 0.0003$, surviving Holm's correction across the six datasets). The question is forecast-encompassing: whether a source adds to a constant (Harvey et al., 1998), and on ECUSTFD the answer is no. The model is not empty there: paired alone against its own modal constant, it is 0.087 more accurate (95% CI 0.026–0.163). The chain already carries the same information, and even a per-item oracle over chain and model does no better than one over chain and constant (0.246 against 0.247). The prior is different: pairing with it beats pairing with the constant on ECUSTFD by 0.148 on ARE and 0.156 on the log error, both intervals excluding zero. A constant reproduces the model's contribution, not the prior's.

H.3 Ablations

Twenty-four ablations support the results in Sec. 5. Table 52 indexes them all, indicating what each varies, how many configurations, fits, or predictions were scored for it, and what it established; the counts are taken from the released artefacts rather than transcribed. One is reported in full below, the comparison of this paper's rules against the published dynamic-selection schemes and two trained combiners (Sec. H.3.1), because it bears on what the fusion result means. Four more have sections of their own: the audit of the metric-scale protocol itself (App. D.6), the direct check of the depth maps and masks against the ground truth for the four datasets that ship it (App. D.7), the cue ablation that explains why the oracle gap is not closed (App. H.7), and the threshold sweep that tests whether the validity criteria were chosen to produce the verdicts we report (App. C.7). We report the constant-source control that bears on them as strongly with the results it audits (Sec. 5.4), and the fold-grouping and pool-size sweeps are in Apps. E.6 and H.6.

Five ablations returned results that support the reading most favourable to this paper: no cue set recovers the oracle gap; depth precision does not order volume accuracy; mask quality accounts for

very little of the volume error that follows it; source diversity does not predict the realised fusion gain; and no prompt form converts the semantic channel into a metric one. We report them at the same length as the rest.

Table 52: **Every ablation in this paper.** What each varies, how many scored configurations, what it settled, and where the detail is. Counts are read from the released artefacts, so the index cannot drift from what was run.

Ablation	What varies	Scored	What it settled	Where
Scale protocol	one recovery rule against fitted constants	31,531	four backbones land on one scene depth wherever a physical scale cue exists	App. D.6
Depth against the sensor	predicted vs shipped depth, per image	7,536	most of the depth error is scale, and better depth is not better volume	App. D.7
Masks against the annotation	predicted vs shipped masks, per object	12,413	silhouettes are accurate, their boundaries much less so	App. D.7
Control sources	constant, global prior vs the VLM	12	the gain is shrinkage; no source separable from a constant	Sec. 5.4
Fold grouping	rows vs target instances	5	row grouping inflates the prior up to $4.4\times$	App. E.6
Fold count	3, 5 and 10 grouped folds	18	the winning side never changes; its margin does	App. E.7
Fold assignment	50 random target partitions	300	the comparison does not rest on a lucky partition	App. E.7
Validity thresholds	TARGETS, SPREAD, the correlation floor	6	the family shrinks to ECUSTFD at 50 targets; its verdict holds	App. C.7
Pool size	best pool at each $k \in [1, 5]$	30	the gain arrives at $k{=}2$ and never grows	App. H.6
Selector cues	five cue groups against a shuffled control	30	no cue set recovers the oracle gap	App. H.7
Injected conflict	one mis-scaled source, five biases	2,280	rank rules degrade smoothly, the mean does not	Sec. H.11
Leave-one-source-out	each source removed in turn	1,908	2–14% per source, one case at $+1316\%$	Sec. H.11
Prompt form	five prompts and crops for the VLM	129	every prompt that moves the correlation leaves it under $r^2{=}3\%$	App. G.4
Panel scale	twelve models in fourteen runs, 2B–235B	106	no scale resolves the instance where the benchmark cannot	Sec. 5.3
Source diversity	error correlation against realised gain	76	diverse pools gain more, but weakly: r^2 near an eighth	App. H.4
Split protocol for abstention	rows shuffled vs targets held out	75,518	shuffled AUROC 0.98 against 0.84–0.93 grouped	Sec. E.5
Confirmatory family	three members at $0.05/3$, YCB post hoc	4	ECUSTFD survives, MADIMA23 fails, YCB holds post hoc	Sec. 5.2
Support-pool controls	constants on the full support sets	10	a constant beats both fused support pools	Sec. 5.4
Dynamic selection	three published schemes, two combiners	608	a selector given the sources' outputs beats the nested rule on 26 of 46	Sec. H.3.1
Selector under conflict	two selectors at five biases	150	the regression selector moves from 1.00 to 1.02 at $4\times$	Sec. H.11
Geometry on Nutrition5k	sensor arm and four backbones	39	resolves instances; calibrated, adds one point to the semantic pair	Sec. G.6
Who supplies the label	a predicted name; a training-split fit	8	the prior survives a predicted label where recognition holds	Sec. 5.2
Metric asymmetry	every verdict under the log error	6	the prior wins on all six, MetaFood3D included	Sec. E.8
Conflict member	the duplicated member: best, first, worst	90	the ranking of rules does not depend on the choice	Sec. H.11

H.3.1 PUBLISHED DYNAMIC-SELECTION SCHEMES AND TRAINED COMBINERS ON THE SAME POOLS

The learned selector of Sec. 3.1 is a random forest, chosen for convenience rather than as a representative of the dynamic-selection literature. Table 53 therefore scores three published schemes on the same 46 pools, in regression form and under the same instance-grouped folds; dynamic integration of regression models has a literature of its own (Rooney et al., 2004; Mendes-Moreira et al., 2012), whose schemes we do not reproduce. OLA (Woods et al., 1997) takes the source with the lowest error on the k nearest training items in the standardised cue space. KNORA-E and KNORA-U (Ko et al., 2008) respectively average the sources that are within a tolerance in ARE on all of those neighbours, shrinking the neighbourhood until one qualifies, and weight every source by the number of neighbours it is within that tolerance on. The neighbourhood size $k \in \{5, 7, 11\}$ and the tolerance, from 0.1 to 2.0, are chosen on each training fold by an inner grouped split that maximises KNORA-E's accuracy, and the chosen pair is shared with OLA and KNORA-U rather than tuned for each separately. Because the folds are grouped by target instance, a neighbour is never another view of the test item. META-DES (Cruz et al., 2018) is not run, since its meta-features are defined on classifier outputs. Two trained combiners are added, because the untrained rules of Sec. 3.1 contain

no blend fitted to the truth: a precision-weighted mean in the classical sense, each source weighted by the inverse variance of its log residual on the training fold once its systematic part is removed, and a linear stack, least squares of the truth on the sources' predictions with an intercept, fitted on the training fold and applied to the held-out one.

Counting how often a scheme improves on the best single source says nothing about whether the schemes differ from one another, so the comparison this field would expect is also run: a Friedman test on the ranks of the nineteen rules, with a Holm-corrected Wilcoxon of each against the best-ranked. It is run on the 36 pools that hold three or more sources, not on all 76; 18 of them are the panel ensembles of Sec. 5.3, which Table 53 does not show, because on a two-source pool the mean, the median, the trimmed mean and the conflict-aware rule are the same estimator (App. C.2) and ranking them there as four competitors would inflate the statistic. The rules are not interchangeable ($\chi^2(18) = 143.9$, $p = 1 \times 10^{-21}$, nineteen rules). The pools share sources and come from six datasets, exactly as in the diversity relation of Sec. 5.4, so the level is optimistic, and we read the ranking rather than the p-value. The best mean rank is the regression selector given the sources' own outputs (5.86), and after correction five schemes are not distinguishable from it: the log-space stack (6.22, $p = 0.66$), the paper's own nested-selected rule (6.40, $p = 0.48$), nested selection with the linear stack admitted (7.17, $p = 0.66$), the classical precision-weighted mean (7.50, $p = 0.48$) and the cubic-centimetre stack (11.31, $p = 0.12$). Everything else is distinguishable, down to the plain mean at 15.71, with KNORA-U and the trimmed mean at 13.26 and 12.58 and KNORA-E at 11.51. The more honest reading is that five schemes share the top, rather than two: this field offers a band of comparable methods on these pools, not a winner. Two readings follow. The nested selection the main text reports is at the top of this field and not distinguishable from the best scheme we found, which is more than the improvement counts alone establish. And what shares that top is a selector reading the sources' own disagreement and a nested rule allowed to blend, both of which the main text calls the open problem.

The three schemes are the classification-derived rules run in regression form with an error tolerance; regression-specific dynamic-selection schemes are not run, so this comparison is indicative rather than exhaustive. OLA improves on the best single source in 13 of the 46 pools, against 20 for the nested-selected rule, and worsens it in 24; head-to-head, and under the same 1% tolerance, it beats the nested-selected rule in 11 pools and loses in 25. On the three pools where the learned gate realises most, the two support pools and the ECUSTFD viewpoint pool, OLA lands close to it (4.86 against 4.88 on YCB, 0.93 against 0.84 on NOCS, 1.69 against 1.66 on the viewpoint pool), so the gains reported in Sec. 5.4 are a property of the cue space rather than of the forest. The two ensemble variants are worse: KNORA-E improves 15 pools and worsens 26, KNORA-U improves 12 and worsens 31, because averaging the qualifying sources reintroduces the gross overestimates that selection removes (12.46 and 9.66 on the YCB support pool). The classical precision-weighted mean is the surprise of this table: it improves 24 pools and worsens 21; more pools improved than the paper's own nested selection manages. We report it as it stands rather than explain it away. Weighting each source by the inverse variance of its training-fold relative error is the one untrained-in-spirit rule here that uses the truth, and on these pools that is worth more than the per-item cue space the selector reads; it remains a mean, so it inherits the gross over-estimates, which is why its rank (7.50) sits below the selectors it beats on counts. The linear stack in cubic centimetres improves 20 pools and worsens 25, beating the nested-selected rule on 20 and losing on 25; refitted in log space it improves 33 and worsens 13 under that same tolerance (34 and 12 without it, as Table 50 reports), the best count in the table; on the three showcase pools it does best of all, 0.84 on the ECUSTFD viewpoint pool against the selector's 1.66 (below the selection oracle's 1.49, which a blend is not bound by), 4.39 against 4.88 on the YCB support pool and 0.73 against 0.84 on NOCS. It is as volatile as the selector elsewhere. Three further columns, set apart in Table 54, answer three objections to the protocol. First, the best single source of Table 49 is identified on the full pool; chosen on the training fold instead and applied out of fold, it coincides with the full-pool choice in 38 of the 46 pools, is worse by more than one point in 8 (the YCB support pool most, where the choice flips between the two segmenters across folds), and the nested-selected rule beats it in 20 pools and loses in 17, the same counts as against the full-pool best. Second, admitting the stack as a candidate inside the nested selection improves 25 pools and worsens 14, a median captured headroom of 17% against the paper's zero, and it beats the paper's nested rule on 10 pools while losing on 10, so on a given pool it is no safer. Third, and most consequential, the regression selector given the sources' own log outputs and their log-spread as cues beats the nested-selected rule on 26 pools and loses on 11, improves on the best single source in 24 and worsens it in 14, and captures

a median 11% of the headroom; on the NOCS support pool it reaches 0.63 against the gate's 0.84. Part of the competence signal that App. J.3 calls missing is therefore in the sources' disagreement, which the pre-integration cues withhold; the remainder, the distance to the oracle, no scheme here recovers.

Table 53: **Published dynamic-selection schemes and trained combiners on the same pools.** Mean ARE per pool for the paper's nested-selected rule (Table 49) and for three classical schemes in regression form, with the neighbourhood size and the competence tolerance chosen on the training fold by an inner grouped split, over the same grids for every pool, so a competitor is tuned the way the paper's own rule is: OLA (Woods et al., 1997) takes the best source on the neighbours; KNORA-E and KNORA-U (Ko et al., 2008) keep, or weight, those within tolerance on them. Then the trained combiners: a precision-weighted mean, whose weights are the inverse variance of each source's log residual once its systematic part is removed; the same linear stack fitted in cubic centimetres and in log space; and the intercept alone, a constant fitted with no source input, which is the control of Sec. 5.4 applied to the combiners themselves. Bold, at least 1% below the best single source; §, fails TARGETS. The last row counts pools improved (Sec. H.3.1); the protocol checks are in Table 54.

Dataset	Channel axis	Best single↓	Selected rule↓	OLA↓	KNORA-E↓	KNORA-U↓	Precision weighted↓	Stack (cm^3)↓	Stack (log)↓	Fitted constant↓	Oracle ceiling↓
ECUSTFD	All channels	0.163	0.182	0.183	0.214	1.506	0.171	1.203	0.175	0.618	0.087
ECUSTFD	Depth backbone	1.417	7.069	6.535	6.806	4.006	**1.270**	2.832	**0.537**	**0.665**	0.528
ECUSTFD	Geometry + prior	0.159	0.159	0.173	0.183	1.807	0.162	1.048	0.168	0.665	0.122
ECUSTFD	Geometry + VLM	0.483	**0.359**	6.695	3.533	3.534	0.499	2.969	0.520	0.618	0.246
ECUSTFD	Geometry + VLM + prior	0.163	0.190	0.183	0.183	0.614	0.173	1.250	0.168	0.618	0.108
ECUSTFD	Integration operator	6.493	6.495	6.493	6.497	6.669	**3.703**	**1.197**	**0.536**	**0.665**	6.474
ECUSTFD	Viewpoint	1.732	**1.658**	**1.690**	5.083	1.856	**0.499**	**0.842**	**0.537**	**0.644**	1.486
ECUSTFD	VLM + prior	0.163	0.163	0.163	0.163	0.163	0.173	0.179	0.168	0.618	0.133
MADIMA23	All channels	0.445	**0.324**	0.541	**0.432**	**0.411**	**0.356**	0.563	**0.437**	0.826	0.154
MADIMA23	Depth backbone	0.611	**0.567**	**0.563**	0.626	**0.591**	0.644	**0.502**	**0.424**	0.735	0.440
MADIMA23	Depth sensor	0.607	0.762	0.742	0.629	0.729	0.771	**0.505**	**0.401**	0.745	0.473
MADIMA23	Geometry + prior	0.436	**0.354**	0.459	**0.362**	**0.391**	**0.394**	0.473	**0.420**	0.735	0.317
MADIMA23	Geometry + VLM	0.493	**0.371**	**0.455**	**0.467**	**0.389**	**0.435**	**0.344**	**0.370**	0.767	0.408
MADIMA23	Geometry + VLM + prior	0.445	**0.349**	0.462	0.441	**0.349**	**0.381**	**0.383**	**0.386**	0.767	0.252
MADIMA23	Integration operator	0.611	**0.496**	**0.546**	**0.474**	**0.541**	**0.603**	**0.463**	**0.421**	0.735	0.404
MADIMA23	VLM + prior	0.445	0.445	0.445	0.711	0.500	0.488	0.791	0.553	0.767	0.378
MRGBD§	All channels	0.117	**0.085**	0.149	**0.085**	**0.085**	**0.075**	0.142	0.123	0.187	0.074
MRGBD§	Depth backbone	0.162	**0.133**	0.228	**0.133**	**0.133**	**0.097**	**0.100**	**0.090**	0.187	0.125
MRGBD§	Geometry + prior	0.117	0.142	0.137	0.122	0.175	**0.093**	**0.095**	**0.079**	0.187	0.101
MRGBD§	Geometry + VLM	0.350	**0.333**	**0.340**	**0.315**	0.387	**0.101**	**0.112**	**0.098**	**0.187**	0.256
MRGBD§	Geometry + VLM + prior	0.117	0.117	0.132	0.146	0.179	**0.078**	**0.106**	**0.083**	0.187	0.100
MRGBD§	Integration operator	0.350	0.350	0.350	0.350	0.350	**0.094**	**0.109**	**0.100**	**0.187**	0.350
MRGBD§	VLM + prior	0.117	0.117	0.117	0.117	0.117	0.116	0.117	**0.108**	0.187	0.107
MetaFood3D	All channels	1.597	**1.074**	**1.031**	2.984	4.788	**0.959**	**1.125**	**0.844**	2.648	0.451
MetaFood3D	Depth backbone	1.597	1.712	1.610	**1.447**	5.592	**1.043**	**1.422**	**0.730**	2.648	1.084
MetaFood3D	Geometry + prior	1.597	**1.041**	**1.086**	**1.276**	**1.480**	1.859	1.927	**1.375**	2.648	0.858
MetaFood3D	Geometry + VLM	1.597	1.986	**1.124**	**1.495**	2.179	2.003	2.902	1.828	2.648	0.931
MetaFood3D	Geometry + VLM + prior	1.597	**1.187**	**1.075**	**1.496**	2.078	1.788	2.021	**1.427**	2.648	0.768
MetaFood3D	Integration operator	1.597	1.707	1.727	2.779	2.465	2.779	2.707	1.614	2.648	1.395
MetaFood3D	VLM + prior	1.673	1.673	1.673	2.073	2.966	1.874	1.852	**1.313**	2.648	1.577
YCB	All channels	1.862	2.384	3.818	**1.757**	1.892	2.664	3.891	**1.826**	4.704	0.449
YCB	Depth backbone	13.474	14.001	13.365	18.771	18.539	21.980	**5.136**	**2.396**	**4.600**	11.196
YCB	Geometry + prior	1.854	2.103	**1.691**	2.531	**1.787**	2.072	4.334	2.243	4.600	1.110
YCB	Geometry + VLM	3.404	**1.873**	3.787	3.714	**2.821**	**2.818**	5.252	**2.098**	4.704	0.938
YCB	Geometry + VLM + prior	1.862	**1.469**	3.935	2.412	2.273	1.983	3.943	**1.768**	4.704	0.738
YCB	Integration operator	17.224	18.047	17.167	21.466	21.896	**11.107**	**4.687**	**2.343**	**4.600**	15.023
YCB	Segmentation source	17.224	**4.883**	**4.857**	**12.462**	**9.655**	**10.815**	**4.389**	**2.141**	**4.600**	4.621
YCB	VLM + prior	1.862	**1.507**	**1.770**	2.503	2.503	2.114	3.940	**1.718**	4.704	0.952
NOCS§	All channels	0.503	0.515	0.582	0.609	0.869	0.515	0.646	0.543	0.723	0.172
NOCS§	Depth backbone	1.922	2.001	2.829	2.328	2.323	2.350	**0.734**	**0.592**	**0.735**	1.461
NOCS§	Geometry + prior	0.521	0.521	0.558	0.534	0.554	**0.446**	0.639	**0.479**	0.735	0.308
NOCS§	Geometry + VLM	0.681	0.773	0.819	0.719	0.731	0.738	0.730	**0.590**	0.723	0.414
NOCS§	Geometry + VLM + prior	0.503	0.575	0.605	0.524	0.520	**0.400**	0.646	0.521	0.723	0.224
NOCS§	Integration operator	3.287	3.492	3.526	3.669	3.745	**1.647**	**0.735**	**0.597**	**0.735**	3.079
NOCS§	Segmentation source	3.287	**0.844**	**0.925**	**2.108**	4.765	**1.265**	**0.732**	**0.596**	**0.735**	0.519
NOCS§	VLM + prior	0.503	**0.466**	0.673	**0.468**	0.623	**0.462**	0.646	0.552	0.723	0.252
Pools improved		–	20/46	13/46	15/46	12/46	24/46	20/46	33/46	11/46	–

H.4 SOURCE DIVERSITY AGAINST FUSION BENEFIT

Fig. 12 presents the two panels which underpin the diversity discussion in Sec. 5.4. Both panels regress on the pool's mean pairwise error correlation, whose complement is diversity. Error correlation is weakly and negatively related to both the oracle ceiling and the gain achieved by the selected rule, so diversity is weakly and positively related to both: the pools that benefit most are the most decorrelated, with the support pools at that end of the axis. The association is weak enough (r^2 near an eighth) that it does not predict which pools will gain. We show it because the classical precondi-

Table 54: **Three objections to the protocol, answered on the same pools.** The best single source chosen out of fold rather than on the full pool; nested selection with the linear stack admitted as a candidate; and the selector given the sources' own outputs as cues. The nested-selected rule of Table 53 is repeated for reference. Bold, at least 1% below the best single source; §, fails TARGETS.

Dataset	Channel axis	Best single↓	Selected rule↓	Nested best single↓	Nested sel. + stack↓	Selector + source cues↓	Oracle ceiling↓
ECUSTFD	All channels	0.163	0.182	0.163	1.206	0.182	0.087
ECUSTFD	Depth backbone	1.417	7.069	6.947	7.069	6.488	0.528
ECUSTFD	Geometry + prior	0.159	0.159	0.159	1.040	0.166	0.122
ECUSTFD	Geometry + VLM	0.483	**0.359**	6.784	**0.359**	**0.330**	0.246
ECUSTFD	Geometry + VLM + prior	0.163	0.190	0.163	1.254	0.179	0.108
ECUSTFD	Integration operator	6.493	6.495	6.493	6.514	6.511	6.474
ECUSTFD	Viewpoint	1.732	**1.658**	1.732	**0.842**	**1.552**	1.486
ECUSTFD	VLM + prior	0.163	0.163	0.163	0.163	0.204	0.133
MADIMA23	All channels	0.445	**0.324**	0.445	**0.324**	**0.404**	0.154
MADIMA23	Depth backbone	0.611	**0.567**	0.637	**0.583**	**0.523**	0.440
MADIMA23	Depth sensor	0.607	0.762	0.607	0.627	0.692	0.473
MADIMA23	Geometry + prior	0.436	**0.354**	0.436	**0.354**	**0.431**	0.317
MADIMA23	Geometry + VLM	0.493	**0.371**	0.493	**0.370**	0.496	0.408
MADIMA23	Geometry + VLM + prior	0.445	**0.349**	0.445	**0.377**	0.447	0.252
MADIMA23	Integration operator	0.611	**0.496**	0.611	**0.490**	**0.519**	0.404
MADIMA23	VLM + prior	0.445	0.445	0.445	0.445	0.445	0.378
MRGBD§	All channels	0.117	**0.085**	0.117	**0.085**	0.128	0.074
MRGBD§	Depth backbone	0.162	**0.133**	0.162	**0.105**	0.168	0.125
MRGBD§	Geometry + prior	0.117	0.142	0.117	0.141	0.129	0.101
MRGBD§	Geometry + VLM	0.350	**0.333**	0.509	**0.112**	**0.274**	0.256
MRGBD§	Geometry + VLM + prior	0.117	0.117	0.117	0.116	0.137	0.100
MRGBD§	Integration operator	0.350	0.350	0.350	**0.109**	0.350	0.350
MRGBD§	VLM + prior	0.117	0.117	0.117	0.118	0.117	0.107
MetaFood3D	All channels	1.597	**1.074**	2.186	**1.404**	**0.709**	0.451
MetaFood3D	Depth backbone	1.597	1.712	1.597	1.982	**1.258**	1.084
MetaFood3D	Geometry + prior	1.597	**1.041**	2.186	**1.041**	**0.958**	0.858
MetaFood3D	Geometry + VLM	1.597	1.986	1.597	1.986	1.671	0.931
MetaFood3D	Geometry + VLM + prior	1.597	**1.187**	2.186	**1.187**	**0.958**	0.768
MetaFood3D	Integration operator	1.597	1.707	1.597	1.707	1.695	1.395
MetaFood3D	VLM + prior	1.673	1.673	1.673	1.673	1.673	1.577
YCB	All channels	1.862	2.384	1.862	2.384	**1.440**	0.449
YCB	Depth backbone	13.474	14.001	13.474	**5.136**	13.449	11.196
YCB	Geometry + prior	1.854	2.103	1.854	2.103	**1.769**	1.110
YCB	Geometry + VLM	3.404	**1.873**	3.404	**1.873**	**1.393**	0.938
YCB	Geometry + VLM + prior	1.862	**1.469**	1.862	**1.469**	**1.483**	0.738
YCB	Integration operator	17.224	18.047	17.224	**4.687**	17.534	15.023
YCB	Segmentation source	17.224	**4.883**	24.810	**5.842**	**4.916**	4.621
YCB	VLM + prior	1.862	**1.507**	1.862	**1.507**	**1.233**	0.952
NOCS§	All channels	0.503	0.515	0.503	0.515	**0.357**	0.172
NOCS§	Depth backbone	1.922	2.001	1.922	**0.734**	**1.644**	1.461
NOCS§	Geometry + prior	0.521	0.521	0.521	0.521	0.577	0.308
NOCS§	Geometry + VLM	0.681	0.773	0.681	0.866	**0.575**	0.414
NOCS§	Geometry + VLM + prior	0.503	0.575	0.503	0.575	**0.361**	0.224
NOCS§	Integration operator	3.287	3.492	3.287	**0.735**	**3.199**	3.079
NOCS§	Segmentation source	3.287	**0.844**	3.287	**0.908**	**0.628**	0.519
NOCS§	VLM + prior	0.503	**0.466**	0.503	**0.466**	**0.359**	0.252
Pools improved		–	20/46	0/46	25/46	24/46	–

tion for a fusion gain is low error correlation, so we report whether it holds here; we do not build on it, because the pools share sources, come from six datasets, and the tests are uncorrected.

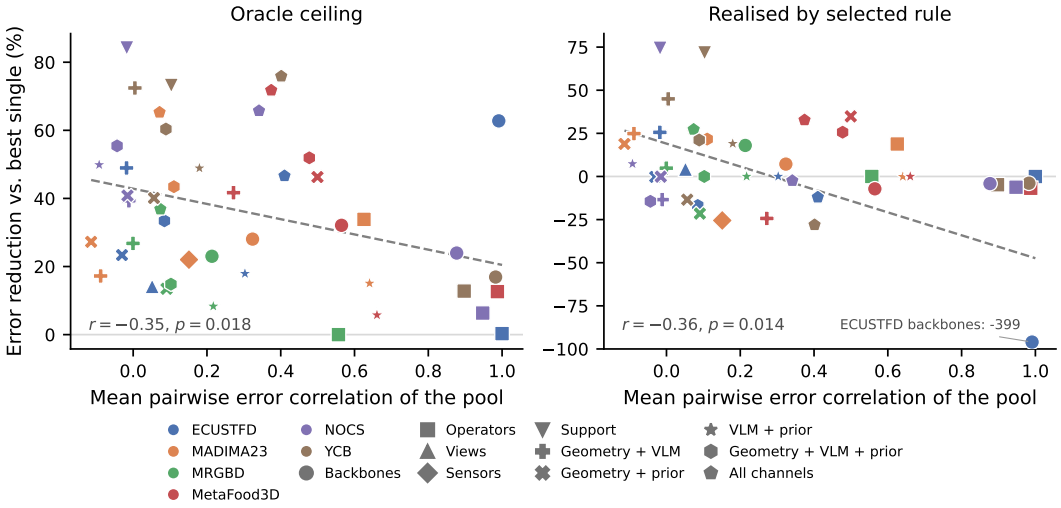

Figure 12: **Source diversity and fusion benefit.** Source diversity versus fusion benefit across the 46 single-axis and heterogeneous pools of Table 49; the panel ensembles and the control pools that complete the 76 are not shown. Both axes are the pool's mean pairwise error correlation, so diversity increases to the left. Left: the oracle selection ceiling decreases with error correlation. Right: so does the gain the selected rule achieves. Marker shape encodes the channel axis; colour encodes the dataset.

H.5 Which channels to combine, and how many

Heterogeneous pools answer which channels are worth combining (the corresponding rows of Table 49). Two results stand out. First, adding the VLM to the category prior improves performance on none of the six datasets the model was scored on. Once recognition is available in tabular form, a 7B VLM adds nothing further. The exceptions are NOCS (0.466 against the prior's 0.503, a 7% gain over fifteen targets the TARGETS criterion leaves unverdicted) and YCB, where the pair reaches 1.51 against the global prior's 1.86 ($H = 39\%$, 95% CI 5–69%) on a dataset that passes TARGETS, and Sec. 5.3 shows that the general pattern is a property of benchmarks whose classes are internally homogeneous rather than of the model: on Nutrition5k the same channel adds up to 0.26 of explained variance beyond the category. Second, geometry adds nothing measurable beyond recognition: paired with the category prior, it does not improve on the prior alone on ECUSTFD or MRGBD, and the pool's best member is the prior. Geometry+VLM, a pool that contains no prior, gains 5–45% over its best member on ECUSTFD, MADIMA23, MRGBD and YCB (and loses 24% on MetaFood3D and 13% on NOCS), but geometry paired with a constant reaches the same error or lower on all four (Sec. G.3); the gain belongs to shrinkage of the chain's gross errors, not to the VLM.

Two channels are the ceiling almost everywhere there is any gain: sweeping every pool of size k drawn from the backbones, the VLM conditions, and the prior, the best pair beats every larger pool on three of the six datasets, ties on two, and is beaten only on MetaFood3D, where the best triple gains, and Sec. H.6 reports the sweep in full. Sec. H.7 explains why the learned selector does not close the oracle gap: when ablated against a shuffled control, the pre-integration cues carry a competence signal too weak to beat the best single source.

H.6 How many channels are worth combining

Table 55 and Fig. 13 show the best pool for each size k, across all combinations of depth backbones, VLM conditions and the category prior. The best pool of a given size is a maximum taken over those candidates on the evaluation set, not a nested selection, so every number in this section is a post hoc optimum and an upper bound on what a method choosing a pool in advance would obtain; the nested figures are the ones in Table 49. k=1 is the best single source, so the first step is as strong as the later ones. The best pair beats the best triple, quadruple, and quintuple on three of the six

datasets and ties on two; on four of the six, the pair combines one depth backbone with the category prior. MetaFood3D alone prefers a triple, and its full quintuple costs 63% more than that triple. The first step varies: the pair improves on the best single by 31% on MADIMA23, 27% on MRGBD, 24% on MetaFood3D, 21% on NOCS, and not at all on ECUSTFD or YCB, where the best single source is already the category or global prior. This is the saturation the ambiguity decomposition predicts: the second source pays off in proportion to how differently it fails, and once a pool holds one geometric and one semantic source, the remaining candidates fail in ways already represented. Here, fusion benefit comes from selecting complementary channels, not accumulating them, which is the opposite of the usual ensembling intuition and why we report pools rather than a single grand ensemble.

Table 55: **Pool-size scaling.** Best mean ARE↓ over all pools of size k from the backbones, VLM conditions and the prior, every k scored on one fixed item set of size n so the columns are comparable.

Dataset	n	k=1	k=2	k=3	k=4	k=5
ECUSTFD	789	0.164	0.163	0.163	0.163	0.164
MADIMA23	40	0.445	0.308	0.325	0.324	0.338
MRGBD	191	0.117	0.085	0.086	0.089	0.096
MetaFood3D	34	1.597	1.213	1.026	1.304	1.673
NOCS	1,477	0.503	0.398	0.401	0.448	0.448
YCB	800	1.854	1.854	1.854	1.854	1.854

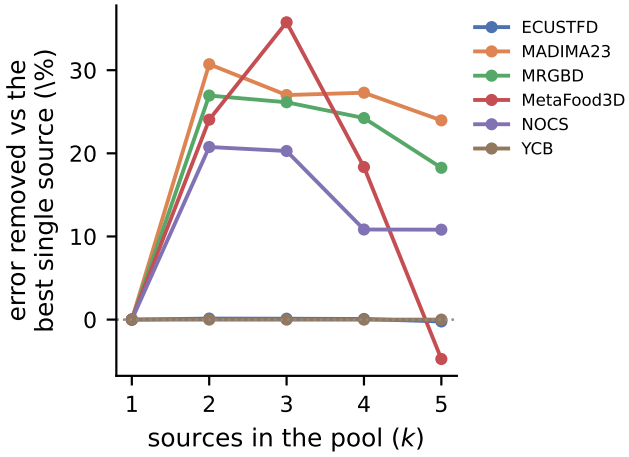

Figure 13: **The fusion gain lies at the second source.** Error against the best single source as sources are added to the best pool of each size. Three datasets peak at k=2 and MetaFood3D at k=3; none go beyond that. ECUSTFD and YCB, whose best single source is already a prior, never improve.

H.7 What the learned selector reads

The oracle gap is large across every valid axis, so the natural question is why a learned selector does not close it. Table 56 ablates the cues it is given against a shuffled control. With all cues, the selector captures a median −10% headroom across the six backbone pools, versus −29% for the shuffled control. Six paired pools with two reversals, and no test, do not establish that the cues carry signal over noise; what they establish is that the real cues are not worse, and that neither set beats the best single source at the median, and any competence they show is confined to two pools (+26% on MADIMA23, +45% on MetaFood3D). These are the regression selector's own out-of-fold gains under grouped cross-validation, which is what Table 56 scores; they are not the realised numbers of Table 49, where the nested procedure may choose an untrained rule on some folds and captures −18% on the MetaFood3D backbone pool. Withholding the depth statistics is the most damaging single change (median −85% and −196% on NOCS), so the selector reads them above all; on MRGBD the shuffled control (−15%) beats the real cues (−183%), so there the cues are actively misleading rather than merely weak. Dropping the segmentation proxies changes nothing, to the last digit, on any dataset, and the reason is mechanical rather than informative: the four segmentation-quality fields are not recorded in the geometric run files, so they enter the cue matrix as a constant column on all six datasets and the forest never had them to drop. The row checks that the ablation removes what it says it removes, not a finding about segmentation quality, and we keep it labelled

as such; on ECUSTFD, every variant, including the control, is dominated by the single degenerate item, which no cue set flags. We report this as a mostly negative result for a reason: the headroom in Table 49 is real, and we have not found a cue set that captures more than a sliver of it.

One cue a reader will reasonably expect to work is the depth backbone's own quality, since that is the number a practitioner reads when choosing one. It does not transfer, for the reasons set out in Sec. D.7: measured directly against each dataset's sensor, depth error does not order volume error either across datasets ($\rho = 0.14$, $p = 0.59$) or within them ($\rho = -0.25$ once configurations are ranked within their own dataset). This matches the negative result from the cue ablation upstream rather than downstream, and together they suggest that the information needed for a fusion rule to pick the right geometric source per item is present in neither the cues nor the backbone's public score.

Table 56: **Cue-group ablation for the learned gate.** Fraction↑ of the best-single-to-oracle headroom the gate recovers when a cue group is withheld, on the depth-backbone pool. The shuffled row destroys the cue–sample alignment and is the control: any variant near it is not using its cues.

Cue variant	ECUSTFD	MADIMA	MRGBD	MF3D	NOCS	YCB
all cues	-575%	26%	-183%	45%	3%	-23%
drop mask/shape	-572%	25%	-112%	18%	10%	0%
drop depth	-577%	18%	-94%	-75%	-196%	-22%
drop segmentation	-575%	26%	-183%	45%	3%	-23%
shuffled (control)	-574%	-21%	-15%	0%	-38%	-153%

H.8 A SOURCE AGAINST ITS OWN WITHIN-CLASS SHUFFLE

Sec. 5.4 asks whether a source contributes per-item information or only per-class location, and answers it by substituting a fixed constant. That control is sound, but it compares against a hand-picked value and requires a target-clustered interval, which two of the six datasets cannot provide. A permutation test answers the same question without either limitation. Under the null that a source contributes its class location and nothing else, its predictions are exchangeable within a class: shuffling which item of a class receives which prediction preserves the source's entire marginal distribution and destroys only per-item signal. We shuffle and refuse each pool $5{,}000$ times.

Only classes holding at least two distinct targets are permuted. A class holding one physical object under twenty views has twenty rows and one ground-truth value, so shuffling inside it moves a prediction between rows asking the same question; counting rows there would report that no source carries instance information when the honest answer is that the test is unavailable. That exclusion removes YCB entirely, where every object is its own class.

The exchangeable unit is the *target*, not the row. Under the null, a source carries its class location and nothing about the individual object, and the many views of one physical target then remain dependent on one another: they are near-identical images, and the source returns near-identical numbers. Shuffling rows would destroy a dependence the null preserves and make the reference distribution too narrow, by a factor of twenty-six on MRGBD, so we permute whole targets within a class. When two targets contribute different numbers of rows, we resample the donor's rows to match the recipient's count; this keeps each target's block intact and keeps the test approximately valid. Each permuted source is fused with the others by the geometric mean, a fixed untrained rule, so that the reference distribution does not also reflect a rule selection.

Table 57 reports 9 of 208 source-in-pool tests below 0.05: 3 of 71 on ECUSTFD, 6 of 68 on NOCS and 0 of 69 on MRGBD. At that level, some ten rejections are what 208 null tests would produce on their own, and none of the nine survives Holm's correction or the Benjamini–Hochberg procedure. The test therefore does not establish per-object information for any source here, and on MRGBD it has little power: with nine distinct targets spread over five classes, exchanging whole targets within a class leaves few distinct permutations, and its p-values there run from 0.053 to 1.0. A permutation test escapes the need for a distributional assumption, not the need for independent units, and nothing in the main text rests on its results.

H.9 FUSING THE DEPTH FIELDS INSTEAD OF THE VOLUMES

Every combination in the main text is decision-level: each backbone runs the whole chain, and the scalar volumes are combined at the end. The channel factorisation of Eq. (1) makes the alternative

Table 57: **A source's contribution against its own within-class shuffle.** Each source is permuted among items of the same class, preserving its whole distribution and destroying only per-item signal, and the pool is refused $5{,}000$ times. *Informative*: sources whose real ordering beats the shuffle at 0.05. Only classes holding at least two distinct targets are permuted, since a class of one target under many views would make the null the identity. *Shrinkage share*: the median fraction of a source's apparent gain the shuffle reproduces.

Dataset	Tests	Classes permuted	Items permuted	Informative	Shrinkage share
ECUSTFD	71	18	820	3	1.05
MRGBD	69	5	191	0	0.98
NOCS	68	5	13,371	6	0.98

obvious, and integration is non-linear, so the volume of an averaged depth field is not the average of the volumes. For each item, we load every backbone's cached depth map, put them on the metric scale that backbone's own chain used, combine them pixel by pixel in log depth, and hand the result to the same integrator with the same mask and field of view. Nothing is refitted, and no constant is introduced, so the comparison isolates where the combination happens.

Table 58 reports both. Where the combination happens does change the answer, and it changes it in opposite directions depending on the rule: on ECUSTFD, fusing depth first is better for the arithmetic mean ($+0.215$ $[+0.060, +0.441]$) and worse for the geometric mean (-0.495 $[-1.409, -0.055]$), while on YCB all three intervals span zero. The first column settles the matter on those two datasets: the best single source reads 1.417 on ECUSTFD and 13.474 on YCB, and neither fusion level approaches it. MRGBD is different, and it is the dataset where the question was most worth asking, since the ceiling of Sec. 5.5 lets a log-space combination reach the prior. Both levels beat the best single backbone (0.162), and the level determines the geometric mean: combined as volumes it collapses to 0.372; combined as depth fields it reaches 0.129, the lowest error in the table ($+0.243$ $[+0.205, +0.274]$). It still does not reach the category prior (0.117), and MRGBD's nine distinct targets fail TARGETS, so we read it as exploratory. The counterfactual therefore corroborates Sec. 5.4 on two datasets and qualifies it on the third: where combination can help at all, where it happens is not a detail.

Three datasets do not enter, for three different reasons. NOCS ships no cached depth. MetaFood3D caches fourteen of twenty-eight maps for one backbone. MADIMA23 has the depth for every image, but several foods share each image, and the cached masks are stacked in the order the segmenter was prompted, so an item cannot be tied to its slice without the segmenter's recorded configuration; guessing would pair one food's depth with another's mask, the error that silently mispaired YCB objects before it was caught.

Table 58: **Where the combination happens changes the answer, and on two of three datasets neither place helps.** The four backbones combined as depth fields before integration, against the same four combined as volumes after it, on identical masks and fields of view with no constant refitted. **Bold**: the paired interval excludes zero. On ECUSTFD and YCB both levels stay behind the best single source. On MRGBD both beat it, and the level decides by how much for the geometric mean; MRGBD's nine targets fail TARGETS, so its intervals are not confirmatory. Not run: NOCS ships no cached depth; MetaFood3D caches fourteen of twenty-eight maps for one backbone; MADIMA23 has the depth, but several foods share each image and the cached masks are stacked in the segmenter's prompt order, so an item cannot be tied to its slice without the segmenter's recorded configuration, and a guess there would pair one food's depth with another's mask.

			Mean ARE↓			
Dataset	Rule	n	Best single	Decision	Feature	Decision−feature
ECUSTFD	Median	837	1.417	2.409	2.162	+0.248 [-0.04, 0.81]
ECUSTFD	Mean	837	1.417	3.521	3.305	**+0.215** [0.06, 0.44]
ECUSTFD	Geometric mean	837	1.417	1.928	2.423	**-0.495** [-1.41, -0.05]
MRGBD	Median	191	0.162	0.151	0.161	**-0.010** [-0.03, -0.00]
MRGBD	Mean	191	0.162	0.133	0.149	-0.017 [-0.02, 0.01]
MRGBD	Geometric mean	191	0.162	0.372	0.129	**+0.243** [0.20, 0.27]
YCB	Median	800	13.474	17.189	17.772	-0.582 [-4.67, 4.38]
YCB	Mean	800	13.474	18.583	18.566	+0.017 [-4.51, 5.51]
YCB	Geometric mean	800	13.474	17.181	18.486	-1.306 [-5.60, 3.81]

H.10 Does the model predict a pool's gain?

For every pool, we estimate each source's log bias and the residual covariance on the training folds, write down the minimum-variance gain in closed form, and compare it with what the pool realises out of fold (Table 59). Over all 76 pools, the prediction is worthless ($r = -0.04$). On the 56 pools that supply at least one distinct target per free parameter of Σ, it predicts the realised gain at $r = +0.83$. Within datasets, the correlation is 0.18 on NOCS, 0.31 on ECUSTFD, and 0.84–0.98 on the other four; the pools share sources, so we give no p-value. The split does not govern harm either. Whether a pool clears it does not predict out-of-fold harm relative to the training-fold best source (Pearson $+0.10$, $p = 0.46$), and the worst failure, an ECUSTFD operator pool that lands at 27,519 against a best member of 6.49, supplies twenty-three targets per parameter. Harm follows a single failure mode instead. In the 60 pools with at least 40 items, where grouped cross-validation can refit the combination, the largest fitted weight correlates with harm at $+0.75$ and -0.14 ($p = 0.30$) without that one pool, whose two sources correlate at 1.000. Shrinking Σ towards a scaled identity (Ledoit & Wolf, 2004) caps its weights and removes the catastrophe. Of the 46 pools with at least one target per parameter among those 60, shrinkage improves 25 and the sample covariance 23 (discordant 3 against 1, exact McNemar $p = 0.62$). The practical rule is to check the conditioning of Σ and shrink when the weights are large; a ratio of targets to parameters says nothing.

Table 59: **Predicted against realised fusion gain.** The prediction uses training folds only: each source's log bias and the residual covariance, then the closed form of Sec. 3.2; the realised gain is that of the same combiner out of fold. The split by distinct targets per free parameter of the covariance is shown because it separates the two groups, not because it governs harm: Sec. 5.5 shows it does not, and that what the large-pool correlation measures is that training-fold moments generalise where pools are large. r and ρ are the Pearson and Spearman correlations between predicted and realised gain, *MAE* their mean absolute difference, and *Sign* the pools whose direction is called correctly.

Pools	n	r	p	ρ	MAE	Sign
All pools	76	-0.04	0.702	0.42	0.280	46/76
≥ 1 **target per parameter**	56	**0.83**	2×10^{-15}	0.82	**0.097**	38/56
< 1 target per parameter	20	-0.29	0.210	-0.17	0.791	8/20

H.11 Behaviour under injected conflict, in full

We inject a mis-scaled duplicate of one member into every pool of three or more sources, at bias $0.25\times$ to $4\times$ (Fig. 14). The mean captures the outlier at once. The median and trimmed mean degrade smoothly, and the conflict-aware rule tracks the median. The dispersion-weighted mean barely moves, but only because it stops listening to its sources and collapses onto the least variable one, which is what makes it catastrophic on ECUSTFD. The learned selectors are the most robust: a source that is wrong by the same factor on every training item is what a selector learns to route around, even though App. H.7 shows it cannot find per-item competence.

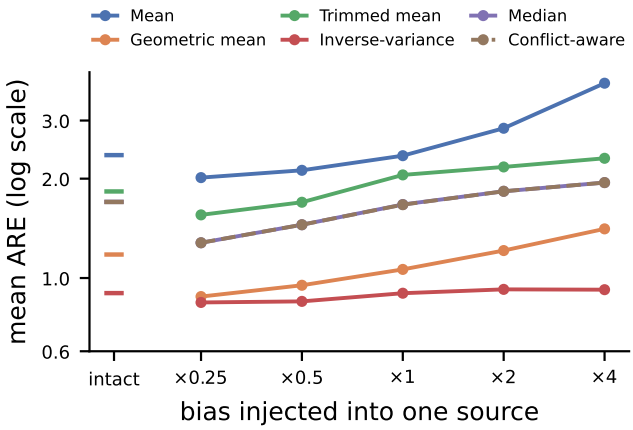

Figure 14: **What each rule answers when one source is badly wrong.** Mean ARE against the bias injected into one member of every pool of three or more sources; dashes mark each rule's intact error. The conflict-aware rule is dashed over the median because the two coincide here.

Table 60 injects a mis-scaled duplicate of one member into every pool of three or more sources and scales its bias from $0.25\times$ to $4\times$; $1\times$ injects the duplicate without bias, and *intact* is the pool before injection. The mean is captured by the outlier and sits at an ARE of 2.36 even intact, because the chain already produces gross overestimates. The median and the trimmed mean degrade smoothly (1.67 to 1.95 and 2.05 to 2.31). The dispersion-weighted mean changes only from 0.90 to 0.92, because it follows whichever source varies least. The conflict-aware rule agrees exactly with the median on 87% of the 75 pool–bias combinations, and across $\tau \in \{0.1, 0.25, 0.5, 1, 2, 4\}$ its mean error moves between 2.82 and 3.00, the default at the bottom. Leaving one source out costs between 2% (the mean) and 14% (the median and the conflict-aware rule) per removed source, with one severe case on ECUSTFD ($+1316\%$) when the removed source is the one the degenerate item does not capture. The regression selector moves from 0.997 intact to 1.022 at $4\times$, and the classification selector from 1.12 to 1.30. Taking the duplicate from the pool's best or worst member instead of its first moves every number and none of the ordering: at $4\times$ the mean reaches 2.63, 3.89 and 5.94 and the median 1.56, 1.95 and 2.69 as the duplicate goes from best to first to worst.

Table 60: **Conflict robustness.** Mean ARE↓ with one badly mis-scaled channel injected into every pool of three or more sources outside YCB; *intact* is the pool before injection, $\times 1$ injects with no added bias. The conflict-aware rule tracks the median throughout. Inverse-variance weighting is lowest in every injected column but reaches it by collapsing onto the least-varying source, the same behaviour that makes it catastrophic intact on ECUSTFD, so the row is not a recommendation. The duplicate is taken from the pool's first source; taking the best or worst member instead moves every number and leaves the ranking (Sec. H.11). The two lower rows are the learned selectors under the same folds, listed apart and not ranked with the untrained rules (Sec. H.11).

Fusion rule	**Intact**	$\times 0.25$	$\times 0.5$	$\times 1$	$\times 2$	$\times 4$
Mean	2.357	2.015	2.120	2.348	2.842	3.892
Geometric mean	1.178	0.879	0.951	1.062	1.211	1.409
Median	1.701	1.279	1.450	1.670	1.831	1.946
Trimmed mean	1.829	1.554	1.697	2.053	2.170	2.305
Inverse-variance	0.900	0.843	0.850	0.899	0.924	0.922
Conflict-aware	1.700	1.279	1.450	1.668	1.831	1.946
Selector (classify)	1.122	1.095	0.970	1.122	1.349	1.297
Selector (regress)	0.997	0.892	0.731	0.997	0.988	1.022

I CLASS-ANCHORED FUSION

This appendix asks whether geometric sources can replace the prior, gives the class-anchored rule in full with its registered confirmation, the multi-source extension and the metric scale as a within-stage fusion, and then the per-source tables behind each.

I.1 CAN A COMBINATION OF GEOMETRIC SOURCES REPLACE THE PRIOR?

The negative results above concern rules that ignore the class. Two questions separate what is unavailable from what was not tried: can any linear combination of the geometric sources *replace* the prior, and does geometry improve on the prior *in combination*?

Table 61 answers the first. For every pool that does not contain the prior, it minimises the mean absolute log error exactly via linear programming over three nested families of bias-corrected combinations fitted to the very items they are scored on, the N-source generalisation of the minimum-variance weights (Bates & Granger, 1969). The comparison favours the families, which are fitted in sample while the prior is scored out-of-fold. On ECUSTFD and NOCS, no pool reaches the prior under any family, even with unconstrained weights and an intercept: within these families there is nothing to find. On MADIMA23, MetaFood3D, and MRGBD, a combination reaches the prior, by more than a constant does. On YCB, only the intercept reaches it, which a constant alone does as well.

I.2 FUSING GEOMETRY ON RECOGNITION'S TERMS, IN FULL

Table 2 answers the second, with the class-anchored rule of Eq. (3). On ECUSTFD, keeping the prior and adding the reference chain's within-class deviation lowers the mean absolute log error by 0.016 (95% CI 0.007–0.026), 0.012 averaged over 30 random fold partitions, all of which it wins; mean ARE falls from 0.159 to 0.146, an interval that touches zero (-0.001 to $+0.026$). Every backbone

Table 61: **How close any linear log-space combination gets to the prior.** For every pool built from geometry and not containing the prior, the mean absolute log error is minimised exactly (a linear program), *in sample*, over three nested families of bias-corrected combinations: weights summing to one, the minimum-variance family; unconstrained weights, which admit shrinkage; and unconstrained weights with an intercept, which is what the log-space stack searches. *Constant*: the best single constant, fitted the same way; it belongs to the intercept family, so where it alone reaches the prior that family's count says nothing about geometry. Fitting on the scored items makes each count a ceiling rather than a method, and the comparison favours the families, whose fit is in sample while the prior is scored out of fold. The panel-ensemble pools are left out: fitted in sample, fourteen weights on 25–53 items reach any target. *Prior*: the out-of-fold prior's mean absolute log error (class median; global median where classes hold one target), per pool, a range where the pools' item sets differ. **Bold**: no pool reaches the prior; §, fails TARGETS.

			Pools whose minimum reaches the prior			
Dataset	**Pools**	**Prior↓**	**Constant**	**Sum to one**	**Free**	**+ intercept**
ECUSTFD	4	0.153–0.154	0 of 4	**0 of 4**	**0 of 4**	**0 of 4**
MADIMA23	4	0.477–0.491	0 of 4	1 of 4	1 of 4	4 of 4
MRGBD§	3	0.121	0 of 3	3 of 3	3 of 3	3 of 3
MetaFood3D	3	0.820	3 of 3	1 of 3	2 of 3	3 of 3
YCB	4	1.012–1.053	4 of 4	**0 of 4**	**0 of 4**	4 of 4
NOCS§	4	0.431–0.471	0 of 4	**0 of 4**	**0 of 4**	**0 of 4**

lowers the prior's log error in all 30 partitions and beats its own values exchanged between whole targets of one class ($p = 0.002$–0.011, 100,000 draws; Table 63), but the interval for Depth Pro touches zero, and across the 16 geometry cells of the datasets that pass TARGETS, the reference chain and MoGe-2 survive Holm's correction ($p = 0.024$ and 0.020) while UniDepthV2 and Depth Pro do not (0.059 and 0.090). The four backbones share the segmenter, scale, and operator, so their agreement is not four independent tests. The gain is small because ECUSTFD's classes barely vary: the model predicts a log-RMSE ratio of 0.94 for the reference chain, and 0.94 is realised.

Full notes to Table 2. The class-anchored rule of Sec. 3.2, fitted on training folds: out-of-fold mean ARE of the prior and of the fused estimate; the gain in mean absolute log error (prior minus fused, target-clustered interval; **bold** excludes zero on a TARGETS-passing dataset); the permutation p against the same source exchanged between whole targets of one class (100,000 draws for the geometry cells of TARGETS-passing datasets, † surviving Holm's correction over those 16 cells; 1,000 otherwise); how many of 30 random target partitions the fused estimate wins, a check of stability across splits of the same targets, not of significance; and the log-RMSE ratio the model predicts from the training-fold within-class correlation against the one realised. Top: each dataset's reference chain. Bottom: the panel member with the median gain (every member in Table 63). Where each class has one target (MetaFood3D, YCB), the prior is the global median, and the rule reduces to a log-linear recalibration of the source. A constant source returns the prior exactly. §, fails TARGETS.

On ECUSTFD, the rule beats every combiner that nests it; on MRGBD, whose nine target groups fail TARGETS, two beat the rule (Table 65). With the same features and folds, a log-space stack of prior and chain, the Mundlak regression that contains both the stack and the rule (Mundlak, 1978), a random-intercept model with empirical-Bayes class effects, and gradient boosting all do worse on ECUSTFD, for every backbone: by 0.014–0.022 in log error for the stack and 0.017–0.029 for the Mundlak regression, with every interval excluding zero, and by 0.04–0.07 for the other two, which fall behind the prior itself. What helps is the constraint. The chain is badly calibrated on ECUSTFD but ranks objects within a class; a rule that takes calibration from the prior can use the ranking without inheriting the calibration error, and a combiner free to re-weight the prior pays for that freedom in variance.

Elsewhere the rule reads as the data allow. On MADIMA23 the reference chain's gain is larger ($+0.088$) but indistinguishable from its within-class shuffle ($p = 0.53$), since two classes with more than one target leave little within-class variation to learn. MRGBD and NOCS give directions ($+0.014$ and $+0.015$). On MetaFood3D and YCB, no test class appears in training, so the rule reduces to a log-linear recalibration of the source around the global median and asks whether geometry ranks objects at all. On YCB it does for all four backbones after Holm's correction ($p \leq 0.013$), but the gains' intervals span zero.

The vision–language channel shows the other regime. On Nutrition5k, all fourteen panel members beat the prior in combination against their within-class shuffles, with 13 intervals excluding zero; the median member lowers mean ARE from 0.545 to 0.388, and the member with the largest gain, Qwen3-VL-235B, from 0.531 to 0.346. On OmniFood8K 13 of 14 do, the median member lowering it from 0.273 to 0.212. On ECUSTFD, no member's gain interval excludes zero.

Fig. 2b sets the predicted log-RMSE ratio against the realised one for the 82 cells with classes to be within. Overall they correlate at $r = 0.87$, but the two mass datasets, with 1,347 and 8,036 targets, carry it: $r = 0.99$ over their 28 cells, against 0.45 over the 54 cells of the volume datasets and 0.24 over the 16 geometry cells. With each dataset's mean removed, $r = 0.75$ (a dataset-resampled interval of 0.33 to 0.98). The formula cannot predict harm, since $\sqrt{1 - r^2_{\text{wc}}} \leq 1$, yet the rule is worse than the prior in 14 of the 54 volume cells, up to 1.30 on MRGBD. The finite-sample condition of Sec. 3.2 accounts for them: 13 of the 14 fail $r^2_{\text{wc}} > 1/n_{\text{eff}}$, and of the 17 volume cells that pass it, one does harm. The model therefore predicts the gain where classes hold many targets and the direction elsewhere, and the condition is the check to run before fusing. Target-grouped conformal intervals at 90% cover 89–93% of targets in most cells, and more (up to 100%) on NOCS and MRGBD, whose few targets make the quantile coarse; the fusion narrows them on MetaFood3D and YCB but not on ECUSTFD (Table 64).

A related question, whether the model predicts a whole pool's gain from its training-fold moments, is answered in App. H.10: the minimum-variance weights predict realised gain at $r = +0.83$ on the 56 pools that supply at least one target per parameter, but that split does not govern harm, which follows a single near-singular pool whose weights shrink towards a scaled identity cap (Ledoit & Wolf, 2004).

What each channel knows beyond the name. The variance counterpart of Eq. (3) is ΔR^2, the log-volume variance a channel explains beyond the category, cross-validated on folds grouped by target and tested against the channel's own values exchanged between whole targets of one class (App. I.9). On ECUSTFD, the category explains 83%. The reference chain adds 0.036 (95% CI 0.012–0.072, permutation $p = 0.001$), about a fifth of what is left within classes, and no VLM adds anything: the incumbent adds -0.002, and no panel member has both a permutation p below 0.05 and an interval above zero. On the two mass datasets, 13 of 14 members do, with medians of 0.21 and 0.26; there the four monocular backbones add 0.06–0.09 on Nutrition5k, as much as its depth sensor (0.08). Geometry carries the instance on both kinds of benchmark; the semantic channel carries it only where classes vary.

I.3 The pre-registered confirmation: what is worth reading

Table 62: **The pre-registered confirmation, on objects and dishes no analysis had seen.** The plan, the samples and these tests were registered before the held-out pipeline ran (App. I.10). Mean absolute log error of the class prior and of the class-anchored rule, their difference with a target-clustered 95% interval (**bold** excludes zero), the one-sided within-class permutation p at 100,000 draws, Holm's correction within each dataset's family, and the log-RMSE ratio the model predicts against the one realised. * the pre-specified primary source of each dataset. ECUSTFD: 604 images of 81 objects disjoint from the paper's; Nutrition5k: 1,738 dishes outside the paper's 1,347, with mass as the target.

		Log error↓					
Sample	**Source**	**Prior**	**Fused**	**Gain [95% CI]**	**Perm. p**	**Holm p**	**RMSE ratio pred. / real.**
ECUSTFD held-out	ZoeDepth*	0.269	0.234	**+0.0343** [+0.0047, +0.0761]	0.016	0.033	0.93 / 0.86
	Depth Pro	0.269	0.232	**+0.0369** [+0.0065, +0.0800]	4×10^{-5}	16×10^{-5}	0.94 / 0.86
	UniDepthV2	0.269	0.261	+0.0080 [-0.0030, +0.0246]	0.267	0.267	0.99 / 0.95
	MoGe-2	0.269	0.252	+0.0163 [-0.0079, +0.0475]	64×10^{-5}	0.0019	0.95 / 0.92
Nutrition5k held-out	ZoeDepth*	0.386	0.376	**+0.0102** [+0.0054, +0.0150]	10^{-5}	5×10^{-5}	0.98 / 0.98
	Depth Pro	0.386	0.377	**+0.0090** [+0.0043, +0.0139]	10^{-5}	5×10^{-5}	0.98 / 0.98
	UniDepthV2	0.386	0.377	**+0.0090** [+0.0044, +0.0137]	10^{-5}	5×10^{-5}	0.98 / 0.98
	MoGe-2	0.386	0.377	**+0.0093** [+0.0046, +0.0140]	10^{-5}	5×10^{-5}	0.98 / 0.98
	RealSense sensor	0.386	0.377	**+0.0088** [+0.0039, +0.0137]	10^{-5}	5×10^{-5}	0.97 / 0.97

Everything above about the class-anchored rule is exploratory: we chose the rule and tuned it after seeing the data they are scored on. We therefore registered the hypotheses, the samples, the tests and the success criteria before running the chain on two samples no analysis in this paper had touched,

and report the outcome whichever way it fell (App. I.10). The first sample is the other half of ECUSTFD: the 81 objects of its `trainval` split, disjoint from the 78 objects behind the 70 targets the paper analyses, 604 side views with a ground-truth volume over 74 targets. The second is the 1,741 Nutrition5k dishes outside the paper's 1,347, labelled by their heaviest ingredient with no dominance threshold, so their classes explain less than the paper's do; 1,738 have images, and the target is mass.

Both primary hypotheses are confirmed (Table 62). A second registration, extending Sec. 5.6, was not. On the held-out ECUSTFD objects, the reference chain's within-class deviation lowers the prior's mean log error by 0.034 (95% CI 0.005–0.076; permutation $p = 0.016$), and three of the four backbones survive Holm's correction. Which backbone leads does not carry over: Depth Pro, which failed the correction on the paper's split, is the strongest here ($p = 4 \times 10^{-5}$), so the rule replicates while the ranking of the sources it fuses does not. On the held-out dishes, every source does, each at the floor of 100,000 draws, with gains of about 0.009–0.010. The paper's headline also replicates on the new objects: the prior beats the chain on 58 to 69 of the 74 targets, depending on the backbone (sign-flip $p \leq 10^{-5}$).

A sensitivity analysis, run after the registered test and labelled as such, shows that the result does not rest on the rule's three free choices. Refitting it on both held-out samples under every setting of the clip (two, three, five robust standard deviations, or none), the floor (0.01 to 5) and the anchor (class median or geometric mean) leaves the gain at $+0.026$ to $+0.037$ on the objects and $+0.0096$ to $+0.0114$ on the dishes, significant in all nine settings on each (Table 68).

Three details are worth reading. The held-out objects are harder than the paper's, and the effect there is larger, not smaller: the prior's log error is 0.269 against 0.154 on the test split, and the gain is 0.034 against 0.016, so the exploratory estimate was not inflated by selection. The model's prediction, which Sec. 5.5 showed holding only loosely on volume benchmarks, is exact on the large held-out sample: predicted ratios of 0.974–0.977 against realised 0.975–0.979 across the five Nutrition5k sources. And the registered comparison against the combiners that nest the rule splits by sample size, as the finite-sample condition implies: on the 74 ECUSTFD targets the rule beats both the log stack and the Mundlak regression, with every one-sided bound above zero, while on 1,738 dishes it still beats the stack but only ties the Mundlak regression, which can afford its free parameters there. The constraint buys protection when targets are few.

I.4 Several sources within class, and the registration that failed

A pool holds more than one source, so the rule extends: keep the prior as the anchor and fit one slope per source on the within-class deviations. Over the 64 pools this rule never significantly loses to the class prior and beats it in 24, and against the single-source rule whose source is chosen inside each training fold it is ahead in 12 pools and behind in 5, the gains sitting on pools that mix channels rather than on variants of one chain. That reading did not survive its own test. We registered it as in Sec. 5.6 and ran it on the top views of the held-out objects: on the pre-specified pool, geometry with a vision–language model, the multi-source rule does not beat the single-source rule (-0.004 in log error, 95% CI -0.016 to $+0.005$, $p = 0.64$), although the operator pool does replicate ($+0.007$, $p = 10^{-4}$). Nor could we find what separates the two: how much the sources' deviations repeat one another predicts the gain weakly over the 45 pools that can be measured (-0.22) and with the wrong sign on the held-out pools. The paper's claim therefore rests on fusing one source's deviation; whether several add anything beyond the best of them is open, and App. I.11 reports the extension in full.

I.5 The metric scale as a within-stage fusion

The scale rule of Sec. 4 is itself a decision-level fusion of cues, and the released scale reports carry every cue's estimate per image; at fixed intrinsics, a volume scales as the cube of the depth scale, so each cue's volume is recoverable without rerunning a network (Table 70). In practice, the released rule is a single cue: its weighted median returns the coin on every ECUSTFD item and the sensor on every MRGBD and YCB item. Precision weights with a per-cue bias, learned on training folds grouped by target, lower the chain's log error from 0.292 to 0.124 on MRGBD, from 1.467 to 0.971 on YCB and from 0.814 to 0.655 on NOCS, but that is not combination. The best single cue times one training-fold factor does as well: 0.099 on MRGBD and 0.627 on NOCS with the sensor, and on YCB the learned rule's lead over the best such cue, 0.099 (95% CI -0.039 to $+0.238$), is not

resolved. What the scale stage lacks is calibration of its one dominant cue, a fitted constant this benchmark's protocol withholds from every chain, not a better way to combine cues.

The same reports test the repair Sec. 5.2 would prescribe for the coin defect. On 16 ECUSTFD items, the coin estimate saturates a clip bound. Letting it abstain and fall through to the support plane makes the chain worse (log error $+0.069$, 95% CI 0.017–0.135), because the plane cue is off by orders of magnitude on this capture. Falling through to the size prior neutralises the degenerate items (mean ARE 6.49 to 0.43) and leaves the log error unchanged. All other numbers in this paper are from the released pipeline (Table 70).

I.6 Class-anchored fusion for every source

Table 63 gives the class-anchored rule of Eq. (3) for all four backbones on the six volume datasets and every panel member on ECUSTFD, MRGBD, NOCS and the two mass datasets. The rule is fitted on the paper's own folds; the permutation null exchanges the source's values between whole targets of one class and refits, $100{,}000$ times for the 16 geometry cells of TARGETS-passing datasets and $1{,}000$ times otherwise; the partitions relabel targets into five folds at random, 30 times. Where every class holds one target, the anchor is the global median and the null exchanges targets across the whole dataset, so it tests whether the source ranks objects at all. Table 64 gives target-grouped conformal intervals at 90%: each target weighted once and left out of its own calibration set. Coverage is close to nominal throughout, and NOCS's 15 targets make its prior over-cover. The fused estimate narrows the interval where it changes the estimate most (MetaFood3D, YCB), and not on ECUSTFD, where its gain is small.

I.7 The class-anchored rule: nesting combiners and the finite-sample condition

Table 65 scores the class-anchored rule beside the combiners that contain it or its parts, on every geometry cell, with the same features and folds. The Mundlak regression, least squares of the log truth on the log prior, the source's clipped within-class deviation and the source's class mean, contains both the log stack (equal coefficients on deviation and class mean) and the rule (coefficient one on the prior, zero on the class mean); the random intercept shrinks class effects by their empirical-Bayes weight and uses the source's level as well as its deviation; gradient boosting sees all four features. On ECUSTFD, the constrained rule is the most accurate for every backbone, ahead of the stack by 0.014–0.022 and of the Mundlak regression by 0.017–0.029 in log error, with every interval excluding zero; the random intercept and the boosted stack fall behind the prior. The nesting models spend their freedom estimating what the prior already fixes. On MRGBD, whose nine target groups fail TARGETS, the stack and the random intercept do better than the rule, and on NOCS nothing beats the prior by much. One-factor recalibration of the source alone is worse than the prior everywhere, except on MRGBD with MoGe-2 (0.099 against 0.121); outside MRGBD, the chain's error is not one scale.

Shrinking the rule's two estimates, and why the paper does not. Both quantities the rule fits are estimated from few targets, so the obvious improvement is to shrink them. We tried both (`fusion/shrunk_anchored.py`). The slope is shrunk by the James–Stein weight for a slope estimated from n_{eff} independent targets, $r_{\text{wc}}^2/(r_{\text{wc}}^2 + 1/n_{\text{eff}})$, which turns the condition of Sec. 3.2 into part of the estimator; the anchor is shrunk towards the global median by $n_c/(n_c + 3)$, the empirical-Bayes weight for a class median estimated from n_c training targets. Neither pays. Shrinking the anchor is harmful: it pulls the class prior towards a global median that is far worse, and the rule then loses to the prior in 46 of the 90 cells instead of 20. Shrinking the slope does what it is designed to do and charges for it: over the 20 cells where the rule loses to the prior it cuts the average damage from 0.0058 to 0.0038 and helps in 19 of them, but over the 70 cells where the rule wins it gives up 0.002, so the mean over all cells is slightly worse than the paper's rule (0.3093 against 0.3082). A shrinkage strength that won would have to be chosen on the same cells it is scored on, which is the practice this paper argues against, so the rule stays as registered, and the condition stays a diagnostic rather than a correction.

The permutation p-values of Table 63 for the 16 geometry cells of TARGETS-passing datasets are recomputed with $100{,}000$ draws, so that a Holm-corrected verdict does not rest on a handful of exceedances. At that resolution, seven of the sixteen cells survive Holm's correction: all four YCB backbones, MetaFood3D with UniDepthV2 (whose gain interval nonetheless spans zero),

Table 63: **The class-anchored rule for every source.** As Table 2, for all four backbones on the six volume datasets and every panel member on ECUSTFD, MRGBD, NOCS and the two mass datasets. §, fails TARGETS.

		Mean ARE↓					
Dataset	**Source**	**Prior**	**Fused**	**Log-error gain [95% CI]**	**Perm. p**	**Partitions won**	**RMSE ratio pred. / real.**
ECUSTFD	geometry, Depth Pro	0.159	0.148	+0.012 [-0.003, +0.027]	0.0112	30/30	0.95 / 0.94
ECUSTFD	geometry, MoGe-2	0.159	0.146	**+0.014** [+0.001, +0.031]	0.0018†	30/30	0.96 / 0.94
ECUSTFD	geometry, UniDepthV2	0.159	0.151	**+0.009** [+0.002, +0.018]	0.0065	30/30	0.99 / 0.97
ECUSTFD	geometry, ZoeDepth	0.159	0.146	**+0.016** [+0.007, +0.026]	0.0024†	30/30	0.94 / 0.94
MADIMA23	geometry, Depth Pro	0.436	0.353	+0.088 [-0.010, +0.183]	0.5263	30/30	0.98 / 0.83
MADIMA23	geometry, MoGe-2	0.436	0.377	+0.046 [-0.043, +0.123]	0.3741	30/30	0.97 / 0.95
MADIMA23	geometry, UniDepthV2	0.436	0.444	-0.001 [-0.033, +0.028]	0.8320	8/30	1.00 / 0.99
MADIMA23	geometry, ZoeDepth	0.436	0.440	-0.011 [-0.033, +0.007]	0.2498	1/30	0.93 / 1.02
MRGBD§	geometry, Depth Pro	0.117	0.113	+0.005 [-0.001, +0.022]	0.311	30/30	0.99 / 0.94
MRGBD§	geometry, MoGe-2	0.117	0.105	+0.014 [+0.005, +0.027]	0.091	30/30	0.93 / 0.87
MRGBD§	geometry, UniDepthV2	0.117	0.116	+0.000 [-0.009, +0.003]	0.415	30/30	0.97 / 1.00
MRGBD§	geometry, ZoeDepth	0.117	0.113	+0.005 [+0.000, +0.013]	0.193	30/30	0.95 / 0.95
MetaFood3D	geometry, Depth Pro	1.673	1.354	+0.019 [-0.175, +0.210]	0.0529	30/30	– / 0.93
MetaFood3D	geometry, MoGe-2	1.673	1.415	-0.005 [-0.204, +0.187]	0.1973	27/30	– / 0.97
MetaFood3D	geometry, UniDepthV2	1.673	1.040	+0.126 [-0.088, +0.344]	0.0004†	30/30	– / 0.82
MetaFood3D	geometry, ZoeDepth	1.673	1.682	+0.001 [-0.100, +0.101]	0.1508	21/30	– / 0.97
YCB	geometry, Depth Pro	1.854	1.820	+0.052 [-0.057, +0.158]	0.0011†	30/30	– / 0.94
YCB	geometry, MoGe-2	1.854	1.807	+0.052 [-0.051, +0.153]	0.0008†	30/30	– / 0.94
YCB	geometry, UniDepthV2	1.854	1.836	+0.069 [-0.034, +0.169]	6e-05†	30/30	– / 0.94
YCB	geometry, ZoeDepth	1.854	1.804	+0.058 [-0.049, +0.165]	0.0004†	30/30	– / 0.93
NOCS§	geometry, Depth Pro	0.521	0.508	+0.009 [-0.009, +0.025]	0.134	30/30	0.98 / 0.99
NOCS§	geometry, MoGe-2	0.521	0.509	+0.008 [-0.013, +0.028]	0.205	30/30	0.98 / 0.99
NOCS§	geometry, UniDepthV2	0.521	0.519	-0.002 [-0.017, +0.013]	0.443	0/30	0.99 / 1.00
NOCS§	geometry, ZoeDepth	0.521	0.499	+0.015 [+0.002, +0.028]	0.099	30/30	0.98 / 0.98
ECUSTFD	Idefics3-8B	0.144	0.141	+0.004 [-0.001, +0.013]	0.036	25/30	1.00 / 0.94
ECUSTFD	InternVL3.5-8B	0.144	0.145	-0.000 [-0.003, +0.002]	0.857	4/30	1.00 / 0.99
ECUSTFD	Llama-3.2-90B-V	0.176	0.182	-0.005 [-0.014, +0.001]	0.299	21/30	1.00 / 1.06
ECUSTFD	LLaVA-OneVision-7B	0.161	0.161	+0.000 [-0.001, +0.001]	0.157	25/30	1.00 / 1.00
ECUSTFD	Molmo2-8B	0.158	0.157	+0.001 [-0.004, +0.006]	0.133	28/30	0.99 / 0.98
ECUSTFD	Qwen2-VL-2B	0.168	0.168	+0.000 [-0.001, +0.001]	0.260	16/30	1.00 / 1.00
ECUSTFD	Qwen2-VL-7B	0.152	0.154	-0.001 [-0.002, +0.000]	0.713	9/30	1.00 / 1.01
ECUSTFD	Qwen3-VL-235B-A22B	0.145	0.145	+0.002 [-0.004, +0.010]	0.523	21/30	0.99 / 0.95
ECUSTFD	Qwen3-VL-30B-A3B	0.147	0.148	+0.002 [-0.003, +0.009]	0.330	21/30	0.99 / 0.96
ECUSTFD	Qwen3-VL-32B	0.146	0.146	+0.002 [-0.002, +0.007]	0.519	18/30	1.00 / 0.98
ECUSTFD	Qwen3-VL-32B (NF4)	0.145	0.145	+0.001 [-0.003, +0.005]	0.361	19/30	1.00 / 0.98
ECUSTFD	Qwen3-VL-8B	0.146	0.142	+0.007 [-0.001, +0.023]	0.026	16/30	1.00 / 0.93
ECUSTFD	Qwen3-VL-8B (NF4)	0.146	0.144	+0.003 [-0.002, +0.011]	0.041	21/30	1.00 / 0.96
MRGBD§	Idefics3-8B	0.117	0.120	-0.004 [-0.015, -0.000]	0.618	0/30	1.00 / 1.06
MRGBD§	InternVL3.5-8B	0.117	0.117	-0.000 [-0.001, +0.000]	0.574	0/30	1.00 / 1.00
MRGBD§	Llama-3.2-90B-V	0.117	0.123	-0.012 [-0.082, +0.005]	0.048	22/30	0.97 / 1.30
MRGBD§	LLaVA-OneVision-7B	0.117	0.115	+0.003 [-0.001, +0.016]	0.829	8/30	1.00 / 0.96

Table 63: **(continued)**

Dataset	Source	Mean ARE↓ Prior	Mean ARE↓ Fused	Log-error gain [95% CI]	Perm. p	Partitions won	RMSE ratio pred. / real.
MRGBD§	Molmo2-8B	0.117	0.118	-0.002 [-0.015, +0.001]	0.405	22/30	0.99 / 1.04
MRGBD§	Qwen2-VL-2B	0.117	0.115	+0.002 [+0.000, +0.005]	0.211	30/30	1.00 / 0.98
MRGBD§	Qwen2-VL-7B	0.117	0.117	+0.001 [+0.000, +0.003]	0.592	27/30	0.99 / 0.99
MRGBD§	Qwen3-VL-235B-A22B	0.117	0.126	-0.011 [-0.038, +0.000]	0.978	0/30	1.00 / 1.14
MRGBD§	Qwen3-VL-30B-A3B	0.117	0.125	-0.012 [-0.063, +0.002]	0.433	22/30	0.98 / 1.22
MRGBD§	Qwen3-VL-32B	0.117	0.129	-0.016 [-0.057, -0.003]	0.879	0/30	1.00 / 1.22
MRGBD§	Qwen3-VL-32B (NF4)	0.117	0.123	-0.008 [-0.037, -0.001]	0.784	0/30	1.00 / 1.13
MRGBD§	Qwen3-VL-8B	0.117	0.114	+0.003 [-0.000, +0.013]	0.557	25/30	1.00 / 0.96
MRGBD§	Qwen3-VL-8B (NF4)	0.117	0.112	+0.005 [-0.003, +0.010]	0.190	30/30	0.99 / 0.99
NOCS§	Idefics3-8B	0.502	0.503	-0.000 [-0.006, +0.005]	0.266	6/30	1.00 / 1.00
NOCS§	InternVL3.5-8B	0.490	0.489	-0.010 [-0.042, +0.022]	0.497	10/30	0.98 / 1.00
NOCS§	Llama-3.2-90B-V	0.502	0.486	+0.006 [-0.021, +0.035]	0.129	14/30	0.94 / 0.96
NOCS§	LLaVA-OneVision-7B	0.509	0.530	-0.015 [-0.042, +0.015]	0.757	16/30	0.97 / 1.01
NOCS§	Molmo2-8B	0.490	0.480	+0.004 [-0.023, +0.030]	0.072	27/30	0.96 / 0.98
NOCS§	Qwen2-VL-2B	0.593	0.591	-0.002 [-0.011, +0.007]	0.534	10/30	0.99 / 1.00
NOCS§	Qwen2-VL-7B	0.538	0.516	+0.013 [-0.015, +0.041]	0.122	27/30	0.97 / 0.97
NOCS§	Qwen3-VL-30B-A3B	0.502	0.411	+0.066 [-0.034, +0.167]	0.019	29/30	0.80 / 0.85
NOCS§	Qwen3-VL-32B	0.490	0.421	+0.051 [-0.015, +0.119]	0.008	30/30	0.86 / 0.86
NOCS§	Qwen3-VL-32B (NF4)	0.490	0.439	+0.038 [-0.025, +0.100]	0.023	30/30	0.87 / 0.89
NOCS§	Qwen3-VL-8B	0.490	0.424	+0.047 [+0.006, +0.087]	0.018	30/30	0.91 / 0.91
NOCS§	Qwen3-VL-8B (NF4)	0.502	0.464	+0.023 [-0.032, +0.081]	0.037	30/30	0.90 / 0.93
Nutrition5k	Idefics3-8B	0.531	0.442	**+0.064** [+0.051, +0.077]	≤ 0.001	30/30	0.87 / 0.86
Nutrition5k	InternVL3.5-8B	0.531	0.389	**+0.103** [+0.088, +0.118]	≤ 0.001	30/30	0.76 / 0.77
Nutrition5k	Llama-3.2-90B-V	0.531	0.362	**+0.125** [+0.109, +0.141]	≤ 0.001	30/30	0.72 / 0.72
Nutrition5k	LLaVA-OneVision-7B	0.531	0.524	**+0.013** [+0.005, +0.021]	≤ 0.001	30/30	0.95 / 0.96
Nutrition5k	Molmo2-8B	0.531	0.433	**+0.068** [+0.054, +0.081]	≤ 0.001	30/30	0.84 / 0.85
Nutrition5k	Qwen2.5-VL-7B	0.545	0.388	**+0.110** [+0.093, +0.127]	≤ 0.001	30/30	0.76 / 0.76
Nutrition5k	Qwen2-VL-2B	0.534	0.511	**+0.015** [+0.006, +0.023]	≤ 0.001	30/30	0.96 / 0.96
Nutrition5k	Qwen2-VL-7B	0.531	0.532	+0.001 [-0.001, +0.004]	0.003	30/30	0.99 / 1.00
Nutrition5k	Qwen3-VL-235B-A22B	0.531	0.346	**+0.138** [+0.121, +0.154]	≤ 0.001	30/30	0.70 / 0.70
Nutrition5k	Qwen3-VL-30B-A3B	0.535	0.345	**+0.137** [+0.120, +0.153]	≤ 0.001	30/30	0.69 / 0.69
Nutrition5k	Qwen3-VL-32B	0.535	0.366	**+0.120** [+0.104, +0.136]	≤ 0.001	30/30	0.73 / 0.73
Nutrition5k	Qwen3-VL-32B (NF4)	0.535	0.372	**+0.115** [+0.099, +0.132]	≤ 0.001	30/30	0.74 / 0.74
Nutrition5k	Qwen3-VL-8B	0.531	0.376	**+0.118** [+0.102, +0.134]	≤ 0.001	30/30	0.74 / 0.75
Nutrition5k	Qwen3-VL-8B (NF4)	0.535	0.377	**+0.121** [+0.105, +0.137]	≤ 0.001	30/30	0.75 / 0.75
OmniFood8K	Idefics3-8B	0.273	0.253	+0.002 [-0.001, +0.005]	≤ 0.001	30/30	0.91 / 0.91
OmniFood8K	InternVL3.5-8B	0.273	0.205	**+0.036** [+0.031, +0.040]	≤ 0.001	30/30	0.73 / 0.75
OmniFood8K	Llama-3.2-90B-V	0.273	0.212	**+0.032** [+0.028, +0.036]	≤ 0.001	30/30	0.75 / 0.76
OmniFood8K	LLaVA-OneVision-7B	0.298	0.287	**+0.004** [+0.001, +0.006]	≤ 0.001	30/30	0.93 / 0.94
OmniFood8K	Molmo2-8B	0.273	0.251	**+0.009** [+0.007, +0.012]	≤ 0.001	30/30	0.92 / 0.91

Table 63: **(continued)**

Dataset	Source	Mean ARE↓ Prior	Mean ARE↓ Fused	Log-error gain [95% CI]	Perm. p	Partitions won	RMSE ratio pred. / real.
OmniFood8K	Qwen2.5-VL-7B	0.263	0.222	**+0.018** [+0.014, +0.022]	≤ 0.001	30/30	0.79 / 0.82
OmniFood8K	Qwen2-VL-2B	0.278	0.259	**+0.005** [+0.003, +0.008]	≤ 0.001	30/30	0.93 / 0.93
OmniFood8K	Qwen2-VL-7B	0.273	0.273	-0.001 [-0.002, -0.001]	1.000	0/30	0.99 / 1.00
OmniFood8K	Qwen3-VL-235B-A22B	0.273	0.199	**+0.040** [+0.036, +0.044]	≤ 0.001	30/30	0.70 / 0.73
OmniFood8K	Qwen3-VL-30B-A3B	0.273	0.195	**+0.044** [+0.040, +0.049]	≤ 0.001	30/30	0.69 / 0.72
OmniFood8K	Qwen3-VL-32B	0.273	0.201	**+0.040** [+0.035, +0.044]	≤ 0.001	30/30	0.71 / 0.73
OmniFood8K	Qwen3-VL-32B (NF4)	0.273	0.200	**+0.042** [+0.038, +0.046]	≤ 0.001	30/30	0.71 / 0.72
OmniFood8K	Qwen3-VL-8B	0.273	0.206	**+0.035** [+0.031, +0.040]	≤ 0.001	30/30	0.72 / 0.75
OmniFood8K	Qwen3-VL-8B (NF4)	0.273	0.206	**+0.037** [+0.033, +0.041]	≤ 0.001	30/30	0.72 / 0.75

Table 64: **Target-grouped conformal intervals at 90%.** Cross-conformal on the out-of-fold absolute log residuals, each target weighted once and excluded from its own calibration set: target-averaged coverage and the median multiplicative half-width, for the prior and for the class-anchored fused estimate. §, fails TARGETS.

			Prior		Fused	
Dataset	**Channel**	**Source**	**Cov.**	**Width**	**Cov.**	**Width**
ECUSTFD	geometry	zoedepth	90%	×1.40	91%	×1.44
ECUSTFD	vlm	Qwen3-VL-8B	90%	×1.37	90%	×1.36
MADIMA23	geometry	depthpro	92%	×3.61	92%	×3.42
MRGBD§	geometry	moge2	89%	×1.46	98%	×1.31
MRGBD§	vlm	Qwen3-VL-8B (NF4)	89%	×1.46	89%	×1.48
MetaFood3D	geometry	zoedepth	91%	×9.53	91%	×7.88
YCB	geometry	zoedepth	90%	×12.58	92%	×8.89
NOCS§	geometry	zoedepth	100%	×2.96	94%	×3.03
NOCS§	vlm	Qwen3-VL-30B-A3B	100%	×2.96	90%	×2.84
Nutrition5k	vlm	Qwen3-VL-235B-A22B	90%	×2.56	90%	×1.93
OmniFood8K	vlm	Qwen3-VL-30B-A3B	90%	×1.64	90%	×1.43

and ECUSTFD with the reference chain and MoGe-2 (Holm-adjusted $p = 0.024$ and 0.020). UniDepthV2 and Depth Pro do not on ECUSTFD (0.059 and 0.090). At $1{,}000$ draws, the reference chain missed (0.081); the difference is Monte Carlo error from counting a few exceedances, which is why the count increased.

The finite-sample condition of Sec. 3.2 is scored on the 82 cells with classes to be within: n_{eff} counts the targets in classes holding at least two, or all targets where the anchor is global. Of the 14 volume-dataset cells where the rule is worse than the prior, 13 fail $r^2_{\text{wc}} > 1/n_{\text{eff}}$ (nine of them on MRGBD, whose n_{eff} is nine); of the 17 volume cells that pass, one does harm, a NOCS panel member at a realised ratio of 1.01. All 28 mass-dataset cells pass, and none does harm. The correlation between predicted and realised ratio is 0.99 on the mass datasets, 0.45 on the volume datasets, 0.50 within ECUSTFD, 0.28 within MRGBD and -0.33 within MADIMA23's four cells.

Table 65: **The class-anchored rule against the combiners that nest it.** Out-of-fold mean absolute log error on each geometry cell, all fitted in log space on the same grouped folds: *log stack*, least squares on the log prior and the log source; *Mundlak*, least squares on the log prior, the source's clipped within-class deviation and its class mean, which contains both the stack and the rule; *random intercept*, the source with empirical-Bayes class effects, fitted on one row per target; *boosted*, gradient boosting on the same four features; *recal.*, the source times one training-fold factor. The prior enters every combiner as the out-of-fold prediction the paper's pools hold. **Bold**: the lowest error among the combiners. * reference chain; §, fails TARGETS.

Dataset	**Backbone**	**Prior**	**Class-anch.**	**Log stack**	**Mundlak**	**Rand. int.**	**Boosted**	**Recal.**
ECUSTFD	ZoeDepth*	0.154	**0.138**	0.152	0.159	0.182	0.190	0.477
ECUSTFD	Depth Pro	0.154	**0.143**	0.160	0.159	0.181	0.204	0.536
ECUSTFD	UniDepthV2	0.154	**0.145**	0.159	0.174	0.197	0.189	0.875
ECUSTFD	MoGe-2	0.154	**0.140**	0.162	0.163	0.178	0.207	0.605
MADIMA23	ZoeDepth	0.485	**0.496**	0.504	0.509	0.505	0.536	0.782
MADIMA23	Depth Pro*	0.485	**0.396**	0.419	0.505	0.430	0.485	0.777
MADIMA23	UniDepthV2	0.485	0.485	0.514	0.507	0.513	**0.433**	1.098
MADIMA23	MoGe-2	0.485	**0.439**	0.478	0.513	0.503	0.523	0.849
MRGBD§	ZoeDepth	0.121	**0.116**	0.131	0.138	0.119	0.122	0.312
MRGBD§	Depth Pro	0.121	0.116	**0.083**	0.096	0.101	0.096	0.145
MRGBD§	UniDepthV2	0.121	0.120	**0.096**	0.114	0.112	0.100	0.263
MRGBD§	MoGe-2*	0.121	0.107	**0.079**	0.101	0.088	0.089	0.099
MetaFood3D	ZoeDepth*	0.820	**0.819**	0.840	0.835	0.849	1.095	1.080
MetaFood3D	Depth Pro	0.820	0.801	0.882	0.857	0.802	**0.763**	1.299
MetaFood3D	UniDepthV2	0.820	**0.694**	0.725	0.834	0.700	0.721	1.414
MetaFood3D	MoGe-2	0.820	**0.824**	0.903	0.859	0.834	0.862	1.406
YCB	ZoeDepth*	1.012	**0.953**	0.974	1.052	1.085	1.257	1.478
YCB	Depth Pro	1.012	**0.960**	0.976	1.028	1.068	1.433	1.258
YCB	UniDepthV2	1.012	**0.943**	0.961	1.027	1.077	1.252	1.276
YCB	MoGe-2	1.012	**0.960**	0.985	1.036	1.073	1.182	1.278
NOCS§	ZoeDepth*	0.431	**0.416**	0.458	0.455	0.555	0.542	0.851
NOCS§	Depth Pro	0.431	**0.422**	0.469	0.484	0.452	0.485	0.836
NOCS§	UniDepthV2	0.431	0.433	0.476	0.566	**0.432**	0.538	0.919
NOCS§	MoGe-2	0.431	**0.422**	0.472	0.497	0.449	0.486	0.890

I.8 THE FUSION CONDITION, SWEPT

The 54 volume cells test the fusion condition only where the benchmarks happen to sit. The two mass datasets hold 1,347 and 8,036 targets and fourteen sources whose within-class correlation runs from about zero to 0.7, so the condition can be swept. For every panel member we drew 100 random subsamples at each of seven sizes, from 12 to 768 targets, refitted the released rule out of fold exactly as in the paper, dropped the draws in which no training fold yields a defined within-class correlation, which happens at the two smallest sizes, and recorded the training-fold r_{wc}, n_{eff} and whether the fused estimate's log RMSE exceeded the prior's. The sizes, the draws and the definition of harm were fixed before the first run; the analysis is exploratory.

Table 66 gives the result over 18,638 refits. Where $r_{\mathrm{wc}}^2 n_{\mathrm{eff}} \leq 1$ the rule harms in 47% of refits, a coin flip with a median ratio of 1.00; where the condition passes it harms in 5.6%. The harm rate falls monotonically across the boundary, from 51.0% in the lowest bin to 31.8% just above one, 9.4% at four to eight, and zero beyond sixty-four. The condition is a first-order threshold and not a guarantee: just above it a third of refits still lose. It is the check to run before fusing, and it says how far above one a practitioner should want to be.

Table 66: **The fusion condition, swept.** The released rule refitted on random subsamples of 12 to 768 targets of Nutrition5k and OmniFood8K, 100 draws per size for each of the 14 panel members (18,638 refits). Harm is a realised log-RMSE ratio of fused to prior above 1. Where the condition $r_{\mathrm{wc}}^2 n_{\mathrm{eff}} > 1$ fails (1,592 refits) the rule harms in 47%; where it passes (17,046), in 5.6%.

$r_{\mathrm{wc}}^2\, n_{\mathrm{eff}}$	**Refits**	**Harm rate**	**Median ratio**
0–0.25	789	51.0%	1.001
0.25–0.5	303	46.9%	0.995
0.5–1	500	41.6%	0.993
1–2	764	31.8%	0.967
2–4	1,598	21.0%	0.917
4–8	2,200	9.4%	0.854
8–16	2,765	6.0%	0.825
16–64	4,090	0.1%	0.786
64+	5,629	0.0%	0.740

I.9 WHAT EACH CHANNEL KNOWS BEYOND THE OBJECT'S NAME

Table 67 reports ΔR^2, the log-volume variance a channel explains beyond the category, cross-validated on folds grouped by target: the category means are fitted on the training folds, and the channel enters only through its within-class deviation, with a slope fitted on the same folds. A channel that is a function of the category contributes nothing. The null for a channel that reads no training labels is the same channel with its values exchanged between whole targets of one class, refitted 1,000 times, as in App. H.8. The fold-refitted priors are printed as a reference, not subtracted: their training rows are computed from the other folds' labels, and the fitted slope turns that leak into explained variance, which is why they score above zero ($+0.021$ on ECUSTFD) although they carry no instance information. A class lookup that is not refitted per fold scores exactly zero. A channel is marked as carrying information when its permutation p is below 0.05 and its bootstrap interval, clustered on targets, lies above zero; both are needed, because on datasets with few targets per class the shuffle can place a small ΔR^2 in its tail while the interval still spans zero (NOCS, Qwen3-VL-8B in NF4: $p = 0.017$, interval -0.015 to 0.290). ΔR^2 is undefined on MetaFood3D and YCB, where no class holds two targets, and is flagged as not estimable where the category-plus-channel fit falls below -1, which happens only when a member returns a handful of extreme values on a small dataset (two MADIMA23 members) or when a global prior is regressed within the five NOCS classes. The NOCS rows scored by the panel are the frames each member was run on (1,376–1,500), not all 13,371, so their category R^2 differs slightly from the geometry rows'.

I.10 THE PRE-REGISTERED CONFIRMATION, IN FULL

Table 62 reports the result; this section gives the plan and what it fixed. The plan behind Sec. 5.6 was written, frozen and published before the chain ran on either held-out sample. It fixed the hypotheses, the two samples and their selection rules, the sources, the outcome, the tests and their draws and seeds, the inference criteria, the exclusion and missing-data rules, a power analysis and the policy

Table 67: **What each channel knows beyond the object's name.** ΔR^2: log-volume variance a channel explains beyond the category, cross-validated on folds grouped by target, with its target-clustered 95% interval, and the permutation p against the same channel exchanged between whole targets of one class (1,000 permutations). **Bold**: the interval excludes zero and $p < 0.05$ on a dataset passing TARGETS. The two priors are fold-refitted and read training labels, so they are a reference, not a null. *Best* is the largest-ΔR^2 member, not the most accurate. §, fails TARGETS.

Channel	n	ARE↓	ΔR^2	95% CI	Perm. p
ECUSTFD (cm^3): category alone explains $R^2 = 0.831$					
geometry	830	6.540	**0.0361**	[0.012, 0.072]	< 0.001
category prior (reference)	837	0.159	0.0210	[-0.011, 0.056]	–
global prior (reference)	837	0.659	0.0006	[-0.014, 0.018]	–
fused (2 operators)	830	6.541	**0.0233**	[0.009, 0.044]	< 0.001
VLM, best: InternVL3.5-8B	1379	3.357	0.0016	[-0.001, 0.005]	0.031
VLM, median of 14 (0 beat the null)	1376	0.801	0.0001	–	–
MADIMA23 (cm^3): category alone explains $R^2 = -0.003$					
geometry	82	0.611	0.0247	[0.013, 0.036]	0.243
category prior (reference)	82	0.436	0.0214	[-0.034, 0.075]	–
global prior (reference)	82	0.457	-0.0651	[-0.186, 0.051]	–
fused (2 operators)	82	0.496	0.0197	[0.008, 0.031]	0.243
VLM, best: Qwen3-VL-8B	40	4.389	0.0199	[-0.012, 0.055]	0.236
VLM, median of 9 (0 beat the null)	40	3.062	-0.0030	–	–
MRGBD§ (cm^3): category alone explains $R^2 = 0.261$					
geometry	191	0.350	0.1500	[0.037, 0.389]	0.040
category prior (reference)	191	0.117	0.2232	[-0.623, 0.949]	–
global prior (reference)	191	0.249	-0.0243	[-0.292, 0.033]	–
fused (2 operators)	191	0.350	0.1500	[0.037, 0.389]	0.040
VLM, best: Llama-3.2-90B-V	191	0.336	0.0965	[0.042, 0.378]	0.059
VLM, median of 14 (0 beat the null)	191	0.371	-0.0019	–	–
NOCS§ (cm^3): category alone explains $R^2 = 0.288$					
geometry	13371	3.287	0.0238	[-0.023, 0.098]	0.197
category prior (reference)	13371	0.521	0.3118	[-0.197, 1.243]	–
fused (2 operators)	13371	3.492	-0.0054	[-0.036, 0.012]	0.480
VLM, best: Qwen3-VL-30B-A3B	1499	0.476	0.2670	[0.041, 0.675]	0.012
VLM, median of 13 (6 beat the null)	1499	0.847	0.0458	–	–
Nutrition5k (g): category alone explains $R^2 = 0.512$					
VLM, best: Qwen3-VL-30B-A3B	1346	0.727	**0.2606**	[0.228, 0.296]	< 0.001
VLM, median of 14 (13 beat the null)	1346	0.784	0.2089	–	–
OmniFood8K (g): category alone explains $R^2 = 0.530$					
VLM, best: Qwen3-VL-30B-A3B	8036	0.464	**0.3052**	[0.285, 0.326]	< 0.001
VLM, median of 14 (13 beat the null)	8036	0.389	0.2599	–	–

for reporting deviations. The frozen code is the repository at tag `prereg-v2`; the plan and the held-out identifier lists, with their checksums, are the release published from that tag, and the hosting service records its publication time. The run script refuses to start until that release's identifier is committed, so no held-out prediction existed when the plan was published. A dry run of the same analysis code on the paper's exploratory samples reproduces its numbers, which is how we checked the code without touching the held-out data.

What was registered as primary. This plan's two hypotheses, each at $\alpha = 0.05$ on its own sample (the multi-source extension of Sec. 5.6 carries a registration of its own, reported in Sec. I.11): that the class-anchored rule has a lower mean absolute log error than the class prior, on ECUSTFD's held-out objects and on the held-out Nutrition5k dishes, each with the reference chain's backbone (ZoeDepth) as the pre-specified source, tested by the one-sided within-class permutation test at $100{,}000$ draws. Both are confirmed: $p = 0.016$ and $p = 10^{-5}$. Under Holm across the two primaries, they remain so.

The secondary hypotheses. We registered each backbone as a family per dataset. On ECUSTFD, Depth Pro ($p = 4 \times 10^{-5}$), MoGe-2 (6×10^{-4}) and the reference chain survive Holm; UniDepthV2 does not (0.27), and its gain is the smallest ($+0.008$). On Nutrition5k, all five sources survive. The registered replication of the paper's headline holds: the prior beats the chain on 58–69 of the 74 held-out targets, with a one-sided sign-flip p of at most 10^{-5} for every backbone. Against the combiners that nest the rule, the registered one-sided bounds put the rule ahead of the log stack in all nine cells ($+0.005$ to $+0.052$) and ahead of the Mundlak regression in the four ECUSTFD cells ($+0.010$ to $+0.030$), while on Nutrition5k the two are level (bounds from -0.0007 to 0.0000). The finite-sample condition $r^2_{\mathrm{wc}} > 1/n_{\mathrm{eff}}$ passes in all nine cells, and all nine gain, which is consistent with it but does not test it: no held-out cell falls on the other side of the condition.

What the held-out samples cost the estimates. Both are harder than the samples the paper analyses. The ECUSTFD prior's mean log error rises from 0.154 to 0.269, and Nutrition5k's from 0.435 to 0.386 in the opposite direction only because the held-out dishes are labelled by their heaviest ingredient without a dominance threshold, which changes the class structure. The gains move with them: on ECUSTFD the rule gains 0.034 against 0.016 on the paper's split, and on Nutrition5k 0.010 against 0.043. The exploratory ECUSTFD estimate was therefore not inflated by selection, and the Nutrition5k gain is smaller where the class is a weaker anchor.

The rule's three free choices. The clip on a source's within-class deviation, the floor on an estimate and the class median as the anchor were all fixed while the exploratory analysis ran, so a reader is entitled to ask what the confirmation owes them. Table 68 refits the rule on both held-out samples under each: the clip at two, three or five robust standard deviations and no clip at all; floors of 0.01, 1 and 5; and the class median against the class geometric mean. The gain survives all nine settings on both samples, at $+0.026$ to $+0.037$ on the held-out objects and $+0.0096$ to $+0.0114$ on the held-out dishes, with every permutation p below 0.05. The registered settings are neither the best nor the worst: a tighter clip would have given a slightly larger gain on Nutrition5k and a looser one a slightly smaller gain on ECUSTFD. This is a sensitivity analysis rather than a registered test, and it is reported alongside the confirmation, which remains registered.

Deviations. Two, both recorded with the plan and neither touching the samples, the tests or the criteria. Nutrition5k distributes one dish's depth file with zero bytes, which stopped the registered-unit rule; the rule now logs and skips unreadable files, and the pipeline treats such a map as missing, as the plan's missing-data rule does. The run script gained a resume guard so it did not repeat the four completed ECUSTFD runs after that stop. Three registered dishes have no images at the source and produce no output from any source: 3 of $1{,}741$, far below the 10% at which the plan calls for a rerun on shared items. Coverage is 604 of 604 items on ECUSTFD and $1{,}738$ of $1{,}741$ on Nutrition5k.

I.11 Fusing several sources within class, pool by pool

Table 69 gives the per-pool numbers behind Sec. 5.6. The multi-source rule fits $\log \hat{V} = \hat{m}_c + \sum_s b_s d_s$, the slopes, by ridge least squares on the training folds' within-class deviations, with the ridge chosen from 10^{-4} to 10 (times the mean diagonal of the design) by a three-way grouped split of those same training rows. Deviations are clipped at three robust standard deviations, as for one source, and a test item whose class the training folds never saw takes the global median and

Table 68: **The confirmed result under the rule's three free choices.** The clip on the within-class deviation, the floor on an estimate and the anchor were fixed while the exploratory analysis ran, so they are swept here on the samples of the confirmation, which were untouched at that point. Mean absolute log error of the prior and of the rule, their difference with a target-clustered 95% interval, and the one-sided within-class permutation p at 10,000 draws. A sensitivity analysis, not a registered test: the registered settings are marked.

		Log error↓			
Sample	**Setting**	**Prior**	**Fused**	**Gain [95% CI]**	p
ECUSTFD held-out	clip at 2 robust SD	0.2686	0.2342	**+0.0344** [+0.0043, +0.0778]	0.017
	clip at 3 (registered)	0.2686	0.2343	**+0.0343** [+0.0044, +0.0773]	0.017
	clip at 5	0.2686	0.2425	**+0.0261** [+0.0046, +0.0515]	0.017
	no clip	0.2686	0.2413	**+0.0272** [+0.0058, +0.0570]	0.017
	floor 0.01	0.2686	0.2368	**+0.0318** [+0.0002, +0.0754]	0.017
	floor 1 (registered)	0.2686	0.2343	**+0.0343** [+0.0044, +0.0773]	0.017
	floor 5	0.2686	0.2333	**+0.0353** [+0.0057, +0.0777]	0.017
	class median (registered)	0.2686	0.2343	**+0.0343** [+0.0044, +0.0773]	0.017
	class geometric mean	0.2326	0.1958	**+0.0367** [+0.0008, +0.0936]	0.017
Nutrition5k held-out	clip at 2 robust SD	0.3863	0.3749	**+0.0114** [+0.0065, +0.0164]	10^{-4}
	clip at 3 (registered)	0.3863	0.3761	**+0.0102** [+0.0054, +0.0150]	10^{-4}
	clip at 5	0.3863	0.3766	**+0.0097** [+0.0050, +0.0143]	10^{-4}
	no clip	0.3863	0.3767	**+0.0096** [+0.0049, +0.0142]	10^{-4}
	floor 0.01	0.3863	0.3761	**+0.0102** [+0.0054, +0.0150]	10^{-4}
	floor 1 (registered)	0.3863	0.3761	**+0.0102** [+0.0054, +0.0150]	10^{-4}
	floor 5	0.3863	0.3761	**+0.0102** [+0.0054, +0.0151]	10^{-4}
	class median (registered)	0.3863	0.3761	**+0.0102** [+0.0054, +0.0150]	10^{-4}
	class geometric mean	0.3849	0.3739	**+0.0110** [+0.0062, +0.0157]	10^{-4}

slopes fitted to deviations from the sources' global means. The empirical-Bayes variant replaces b by $w_c b_c + (1 - w_c)b$ with $w_c = n_c/(n_c + 5)$ and n_c the class's training targets. The baseline is the single-source rule, with its source chosen inside each training fold: choosing it on the rows it is scored on would flatten the baseline, and it did so in an early version of this analysis.

The extension was registered and tested like the single-source rule, on the top views of the 81 held-out ECUSTFD objects: images from a viewpoint the paper's geometric runs do not use, for objects it does not analyse. The pre-specified primary was the geometry-and-VLM pool, the test was a one-sided sign-flip over targets at $\alpha = 0.05$, and the plan was to release tag `prereg-v3`, published before those runs. The primary hypothesis fails: -0.004 in log error (95% CI -0.016 to $+0.005$, $p = 0.64$) on 71 targets. Of the secondaries, the operator pool replicates ($+0.007$, $[+0.003, +0.011]$, $p = 10^{-4}$), the four-backbone pool does not (-0.001, $p = 0.69$; registered as underpowered), and the all-channel pool is positive but unresolved ($+0.010$, $p = 0.12$). Against the class prior, the rule holds on every pool ($+0.037$ to $+0.045$ in mean ARE, three of four intervals excluding zero). The registered descriptive check, the gain against targets per fitted slope, shows no relation here: the pool with the most targets per slope gains nothing and the one with the fewest gains most. One observation is worth recording for later work rather than as a result: the empirical-Bayes variant, which shrinks each class's slope towards the pooled one, is the most accurate rule on all four held-out pools (0.214–0.241 against the pooled rule's 0.233–0.262), the reverse of what the paper's own pools show.

Why some pools gain, and others do not, as far as we can tell: not far. The natural explanation is redundancy. One slope per source should pay where the sources' within-class deviations carry different information and cost where they overlap, so we measured it: for each pool, the mean absolute correlation between the sources' deviations, the largest of those correlations, and the condition number of their correlation matrix. Over the 45 pools with at least two informative sources, the relation runs in the expected direction and is weak: the gain over the single-source rule correlates with the mean deviation correlation at -0.22 (Spearman -0.09) and with the log condition number at -0.36 (Spearman $+0.03$). Split at a mean correlation of 0.3, the barely correlated pools gain $+0.009$ on average and the correlated ones lose 0.005, but the counts of resolved differences are almost the same (4 ahead and 2 behind against 4 and 3). The held-out samples then contradict the story outright: the pool that replicated, the two operators, has deviations correlated at 0.87, while the pool that failed, geometry with a vision–language model, has deviations correlated at 0.15. Nor does the number of targets per fitted slope predict the gain (-0.004, Spearman -0.18). We therefore report

no rule for when fusing several sources within class pays. That is the open question this extension leaves, and it is why the paper's claim rests on the single-source rule.

Two table structures are worth naming. Pools that contain the prior as a *member* are not a clean comparison for either rule: a fold-refitted prior's within-class deviation carries which fold an item sat in, the leak Sec. 5.5 describes for ΔR^2, and the single-source rule can fit a slope to it. Restricted to the 34 pools with no prior member, the multi-source rule leads in 7 and lags in 2. And the gains concentrate where a pool has 40 or more targets: over those twelve pools the mean gain is $+0.013$, with five ahead and none behind.

Table 69: **The multi-source class-anchored rule on every pool.** Out-of-fold mean absolute log error of the class prior; of the single-source rule, its source chosen inside each training fold; of the multi-source rule with one slope per source; and of its empirical-Bayes variant, whose per-class slope is shrunk towards the pooled one. The last column is the multi-source rule's gain over the single-source rule (target-clustered 95% interval; **bold** excludes zero). §, fails TARGETS.

				Log error↓				
Dataset	**Pool**	**Src.**	**Targets**	**Prior**	**Single source**	**Multi source**	**Multi, EB**	**Multi − single [95% CI]**
ECUSTFD	depth backbones	4	70	0.154	0.146	0.146	0.146	+0.0002 [-0.0065, +0.0040]
ECUSTFD	operators	2	70	0.154	0.152	0.148	0.148	**+0.0043** [+0.0021, +0.0079]
ECUSTFD	viewpoints	2	70	0.153	0.148	0.153	0.152	-0.0043 [-0.0179, +0.0027]
ECUSTFD	geometry + VLM	2	66	0.153	0.151	0.147	0.145	**+0.0035** [+0.0015, +0.0066]
ECUSTFD	geometry + prior	2	70	0.154	0.107	0.136	0.130	-0.0287 [-0.0456, -0.0101]
ECUSTFD	VLM + prior	2	66	0.153	0.130	0.137	0.146	-0.0078 [-0.0150, +0.0003]
ECUSTFD	geometry + VLM + prior	3	66	0.153	0.130	0.140	0.137	-0.0107 [-0.0248, +0.0073]
ECUSTFD	geometry + constant	2	66	0.153	0.150	0.148	0.144	**+0.0021** [+0.0013, +0.0029]
ECUSTFD	geometry + global prior	2	66	0.153	0.150	0.148	0.144	**+0.0028** [+0.0015, +0.0046]
ECUSTFD	VLM panel	14	136	0.156	0.156	0.156	0.157	-0.0001 [-0.0007, +0.0005]
ECUSTFD	geometry + VLM panel	15	66	0.181	0.185	0.167	0.163	**+0.0182** [+0.0013, +0.0407]
ECUSTFD	VLM panel + prior	15	66	0.181	0.117	0.147	0.141	-0.0301 [-0.0414, -0.0171]
ECUSTFD	all channels	6	66	0.153	0.130	0.143	0.141	-0.0130 [-0.0265, +0.0040]
MADIMA23	depth backbones	4	36	0.485	0.395	0.438	0.438	-0.0436 [-0.0801, -0.0093]
MADIMA23	operators	2	36	0.485	0.395	0.406	0.406	-0.0114 [-0.0318, +0.0063]
MADIMA23	sensors	2	36	0.477	0.392	0.373	0.373	+0.0193 [-0.0160, +0.0539]
MADIMA23	geometry + VLM	2	34	0.491	0.353	0.376	0.376	-0.0234 [-0.0571, +0.0083]
MADIMA23	geometry + prior	2	36	0.485	0.395	0.399	0.399	-0.0042 [-0.0139, +0.0047]
MADIMA23	VLM + prior	2	34	0.491	0.480	0.477	0.477	+0.0023 [-0.0065, +0.0116]
MADIMA23	geometry + VLM + prior	3	34	0.491	0.353	0.387	0.387	-0.0345 [-0.0579, -0.0117]
MADIMA23	geometry + constant	2	34	0.491	0.353	0.351	0.351	+0.0018 [-0.0007, +0.0044]
MADIMA23	geometry + global prior	2	34	0.491	0.353	0.336	0.336	+0.0170 [-0.0080, +0.0427]
MADIMA23	VLM panel	13	24	0.573	0.580	0.577	0.577	+0.0030 [-0.0985, +0.1006]
MADIMA23	geometry + VLM panel	14	24	0.573	0.312	0.431	0.431	-0.1184 [-0.2151, -0.0239]
MADIMA23	VLM panel + prior	14	24	0.573	0.580	0.575	0.575	+0.0049 [-0.0966, +0.1027]
MADIMA23	all channels	6	33	0.491	0.439	0.439	0.439	+0.0004 [-0.0695, +0.0858]
MetaFood3D	depth backbones	4	34	0.820	0.691	0.628	0.628	+0.0629 [-0.0464, +0.1824]
MetaFood3D	operators	2	34	0.820	0.845	0.851	0.851	-0.0060 [-0.0385, +0.0264]
MetaFood3D	geometry + VLM	2	34	0.820	0.937	0.856	0.856	**+0.0816** [+0.0203, +0.1481]
MetaFood3D	geometry + prior	2	34	0.820	0.830	0.833	0.833	-0.0025 [-0.0275, +0.0220]
MetaFood3D	VLM + prior	2	34	0.820	0.908	0.890	0.890	+0.0185 [-0.0226, +0.0636]
MetaFood3D	geometry + VLM + prior	3	34	0.820	0.937	0.870	0.870	**+0.0670** [+0.0105, +0.1324]
MetaFood3D	geometry + constant	2	34	0.820	0.830	0.831	0.831	-0.0006 [-0.0127, +0.0121]
MetaFood3D	geometry + global prior	2	34	0.820	0.830	0.833	0.833	-0.0025 [-0.0275, +0.0220]
MetaFood3D	VLM panel	13	29	0.887	0.882	0.901	0.901	-0.0181 [-0.1130, +0.0738]
MetaFood3D	geometry + VLM panel	14	29	0.887	0.882	0.920	0.920	-0.0380 [-0.1326, +0.0505]
MetaFood3D	VLM panel + prior	14	29	0.887	0.880	0.902	0.902	-0.0216 [-0.1496, +0.1049]
MetaFood3D	all channels	6	34	0.820	0.691	0.697	0.697	-0.0061 [-0.1512, +0.1316]
YCB	depth backbones	4	40	1.012	0.956	0.955	0.955	+0.0010 [-0.0226, +0.0236]
YCB	operators	2	40	1.012	0.978	0.957	0.957	+0.0210 [-0.0021, +0.0448]
YCB	support masks	2	40	1.012	0.967	0.963	0.963	+0.0045 [-0.0295, +0.0417]
YCB	geometry + VLM	2	40	1.053	0.993	0.937	0.937	**+0.0567** [+0.0020, +0.1119]
YCB	geometry + prior	2	40	1.012	0.967	0.968	0.968	-0.0011 [-0.0206, +0.0184]

I.12 THE SCALE CUES, AND THE COIN-CLAMP REPAIR

Table 70 recomputes the reference chain under every scale cue. The released weighted median returns the coin on every ECUSTFD item and the sensor on every MRGBD and YCB item, because a cue at confidence one outweighs the others combined. NOCS reports its cues once per frame while scaling per object, so its comparison is frame-level. The support-plane cue alone is off by orders of magnitude on ECUSTFD and MRGBD, which is why its assumed height matters where it is the only

Table 69: **(continued)**

Dataset	Pool	Src.	Targets	Log error↓ Prior	Single source	Multi source	Multi, EB	Multi − single [95% CI]
YCB	VLM + prior	2	40	1.053	0.996	0.974	0.974	+0.0224 [-0.0178, +0.0645]
YCB	geometry + VLM + prior	3	40	1.053	0.993	0.933	0.933	**+0.0601** [+0.0043, +0.1148]
YCB	geometry + constant	2	40	1.053	0.964	0.968	0.968	-0.0040 [-0.0304, +0.0227]
YCB	geometry + global prior	2	40	1.053	0.964	0.903	0.903	+0.0610 [-0.0234, +0.1511]
YCB	VLM panel	9	11	1.250	0.897	0.907	0.907	-0.0096 [-0.1537, +0.1092]
YCB	geometry + VLM panel	10	11	1.250	0.897	0.879	0.879	+0.0180 [-0.1111, +0.1267]
YCB	VLM panel + prior	10	11	1.250	0.897	0.905	0.905	-0.0076 [-0.1528, +0.1119]
YCB	all channels	6	40	1.053	0.975	0.916	0.916	**+0.0593** [+0.0069, +0.1126]
NOCS§	depth backbones	4	15	0.431	0.429	0.428	0.429	+0.0010 [-0.0025, +0.0042]
NOCS§	operators	2	15	0.431	0.429	0.428	0.429	**+0.0011** [+0.0004, +0.0018]
NOCS§	support masks	2	15	0.431	0.430	0.431	0.420	-0.0010 [-0.0023, +0.0003]
NOCS§	geometry + VLM	2	15	0.471	0.471	0.462	0.465	+0.0091 [-0.0002, +0.0249]
NOCS§	geometry + prior	2	15	0.431	0.406	0.405	0.398	+0.0006 [-0.0055, +0.0068]
NOCS§	VLM + prior	2	15	0.471	0.477	0.454	0.458	+0.0230 [-0.0055, +0.0576]
NOCS§	geometry + VLM + prior	3	15	0.471	0.471	0.447	0.455	+0.0238 [-0.0170, +0.0686]
NOCS§	geometry + constant	2	15	0.471	0.470	0.468	0.467	**+0.0022** [+0.0009, +0.0037]
NOCS§	geometry + global prior	2	15	0.471	0.481	0.481	0.469	-0.0003 [-0.0158, +0.0132]
NOCS§	VLM panel	13	15	0.519	0.495	0.476	0.461	+0.0194 [-0.0297, +0.0558]
NOCS§	geometry + VLM panel	14	15	0.519	0.495	0.474	0.461	+0.0208 [-0.0301, +0.0569]
NOCS§	VLM panel + prior	14	15	0.519	0.495	0.449	0.439	+0.0460 [-0.0150, +0.0996]
NOCS§	all channels	6	15	0.471	0.471	0.457	0.455	+0.0141 [-0.0366, +0.0626]

cue (MetaFood3D). The learned rule weights each cue by the inverse of its robust residual variance after a per-cue bias, both fitted on training folds grouped by target. It is therefore a fusion fitted to training targets, not a constant fitted to the evaluation target, but it is not the protocol the rest of the paper uses. Its per-cue bias is also most of what it gains. The one-factor rows fit that bias to a single cue and nothing else, and the best of them matches or beats the learned rule on MRGBD and NOCS; on MRGBD and NOCS the learned rule even beats the per-item cue oracle, which only a fitted bias can do. The last column compares the learned rule with the best one-factor cue; the difference is not resolved on any dataset, including YCB's $+0.099$. The coin estimate saturates a clip bound (0.5 or 20 cm) on 16 ECUSTFD items. The released code converts a degenerate estimate into the most extreme admissible one rather than rejecting it; it needs to abstain, and the table shows which fall-through works.

J Extended discussion

This appendix provides the full discussion: what a benchmark can know, a reporting checklist, lessons for fusion research, and the limitations.

J.1 What a benchmark can know

Behind the statistics is a question of measurement: when does a benchmark score tell us about the attribute it aims to measure? Machine learning has begun to treat evaluation as measuring an unobserved construct through an operationalisation (Jacobs & Wallach, 2021). A volume benchmark on which a class median scores as well as a reconstruction still rewards a better geometric pipeline, but it fails discriminant validity: a recognition route reaches the same score, so the benchmark cannot say whether a pipeline's accuracy came from geometry or from recognition. The class-anchored result sharpens this. On ECUSTFD, geometry does carry the construct, and the benchmark lets it show only through the small within-class variation recognition cannot explain. Repetition is not independent confirmation either: twelve views of one apple through one pipeline are worth about four independent rows.

J.2 What the field should report

The results imply a minimum reporting standard for any study that credits a channel with a gain. Three of its items concern what the evidence is counted in. Count distinct targets rather than rows, and cluster every interval on them. State where each configuration's metric scale comes from, and that no scale factor was fitted to the evaluation target. And report σ_w, the design effect and the

Table 70: **Every scale cue, and the coin-clamp repair.** The reference chain's volume recomputed under each cue's scale ($V \propto s^3$ at fixed intrinsics), with mean and median ARE, mean absolute log error, and the log-error gain over the released rule (target-clustered 95% interval). *Learned weights*: log-space precision weights with a per-cue bias, fitted on training folds grouped by target. *One factor*: that estimate times a single training-fold factor, its median log bias. The last column is the learned rule's gain over the best one-factor cue, chosen on the evaluation items, which favours the baseline: what it does not gain there is calibration, not combination. On ECUSTFD the coin estimate saturates a clip bound on 16 items; the repair lets it abstain and falls through to the next cue. §, fails TARGETS.

Dataset	Scale from	ARE	Median ARE	Log error	Gain over released [95% CI]	Gain over best one-factor cue
ECUSTFD	released rule	6.493	0.356	0.477	–	
ECUSTFD	coin alone	6.493	0.356	0.477	+0.000 [+0.000, +0.000]	
ECUSTFD	support plane alone	212	185.188	4.951	-4.474 [-4.677, -4.244]	
ECUSTFD	size prior alone	0.698	0.747	1.365	-0.888 [-0.982, -0.781]	
ECUSTFD	median of cues	0.468	0.350	0.450	+0.027 [+0.005, +0.059]	
ECUSTFD	learned weights	0.378	0.266	0.444	+0.034 [-0.044, +0.103]	+0.033 [-0.043, +0.102]
ECUSTFD	released rule, one factor	6.760	0.365	0.477	+0.000 [-0.006, +0.007]	
ECUSTFD	coin, one factor	6.760	0.365	0.477	+0.000 [-0.006, +0.007]	
ECUSTFD	support plane, one factor	0.541	0.375	0.747	-0.270 [-0.454, -0.109]	
ECUSTFD	size prior, one factor	0.587	0.371	0.495	-0.018 [-0.089, +0.050]	
ECUSTFD	per-item cue oracle	0.382	0.322	0.422	+0.055 [+0.028, +0.090]	
ECUSTFD	coin abstains at a bound, plane	6.552	0.361	0.546	-0.069 [-0.135, -0.017]	
ECUSTFD	coin abstains at a bound, size prior	0.430	0.361	0.486	-0.009 [-0.031, +0.013]	
MRGBD§	released rule	0.350	0.337	0.292	–	
MRGBD§	sensor alone	0.350	0.337	0.292	+0.000 [+0.000, +0.000]	
MRGBD§	support plane alone	11.430	11.036	2.492	-2.200 [-2.222, -2.168]	
MRGBD§	size prior alone	0.397	0.440	0.530	-0.238 [-0.372, -0.040]	
MRGBD§	median of cues	0.350	0.337	0.292	+0.000 [+0.000, +0.000]	
MRGBD§	learned weights	0.127	0.103	0.124	+0.168 [+0.074, +0.200]	-0.024 [-0.039, +0.017]
MRGBD§	released rule, one factor	0.100	0.066	0.099	+0.193 [+0.064, +0.230]	
MRGBD§	sensor, one factor	0.100	0.066	0.099	+0.193 [+0.064, +0.230]	
MRGBD§	support plane, one factor	0.193	0.130	0.183	+0.109 [+0.034, +0.188]	
MRGBD§	size prior, one factor	0.361	0.253	0.314	-0.022 [-0.143, +0.080]	
MRGBD§	per-item cue oracle	0.287	0.292	0.253	+0.039 [+0.004, +0.096]	
YCB	released rule	17.224	0.847	1.467	–	
YCB	sensor alone	17.224	0.847	1.467	+0.000 [+0.000, +0.000]	
YCB	support plane alone	13.734	0.942	2.214	-0.747 [-0.954, -0.530]	
YCB	size prior alone	0.869	0.871	2.055	-0.588 [-1.071, -0.111]	
YCB	median of cues	5.202	0.838	1.559	-0.091 [-0.274, +0.094]	
YCB	learned weights	2.879	0.581	0.971	+0.496 [+0.311, +0.677]	+0.099 [-0.039, +0.238]
YCB	released rule, one factor	15.703	0.838	1.478	-0.011 [-0.033, +0.012]	
YCB	sensor, one factor	15.703	0.838	1.478	-0.011 [-0.033, +0.012]	
YCB	support plane, one factor	46.531	0.945	2.064	-0.597 [-0.724, -0.474]	
YCB	size prior, one factor	2.385	0.745	1.070	+0.397 [+0.119, +0.670]	
YCB	per-item cue oracle	0.730	0.557	0.798	+0.669 [+0.467, +0.891]	
NOCS§	released rule	3.287	0.682	0.814	–	
NOCS§	sensor alone	6.199	1.316	0.954	-0.140 [-0.337, +0.049]	
NOCS§	support plane alone	4.642	0.878	0.992	-0.179 [-0.338, -0.023]	
NOCS§	size prior alone	0.968	0.914	2.459	-1.645 [-1.962, -1.329]	
NOCS§	median of cues	3.891	0.786	0.867	-0.053 [-0.190, +0.082]	
NOCS§	learned weights	2.168	0.496	0.655	+0.159 [+0.001, +0.295]	-0.028 [-0.074, +0.020]
NOCS§	released rule, one factor	2.337	0.643	0.851	-0.038 [-0.143, +0.063]	
NOCS§	sensor, one factor	2.442	0.473	0.627	+0.187 [+0.019, +0.336]	
NOCS§	support plane, one factor	2.426	0.634	0.877	-0.063 [-0.218, +0.076]	
NOCS§	size prior, one factor	2.233	0.617	0.828	-0.014 [-0.125, +0.090]	
NOCS§	per-item cue oracle	1.121	0.636	0.682	+0.132 [-0.019, +0.281]	

number of targets needed to detect the claimed effect (Table 12) before claiming it, since all three can be computed from the data alone, before any method exists.

The others concern what a gain is measured against. Report the global and class priors beside the method, fitted on folds grouped by target, because a channel that does not beat the name of the object has not been shown to measure anything. Replace a fusion partner by a constant and by a within-class shuffle: this is the forecast-encompassing question of whether a source adds to a constant (Harvey et al., 1998), and a gain that survives both controls is information rather than shrinkage. Fuse a source with the prior through its within-class deviation, check $r^2_{\mathrm{wc}} > 1/n_{\mathrm{eff}}$ before expecting a gain, and report what the model predicts besides what is realised. Report the oracle ceiling whenever a fusion or selection method is claimed, together with the space the combiner is fitted in. And where a rule was chosen on the data it is scored on, register the test and repeat it on data held out from every analysis, as Sec. 5.6 does here. All but the scale statement, the oracle and the registration need only the per-item predictions and labels, and the released scripts compute them for any benchmark in the same format.

J.3 Significance for fusion research

Three lessons carry beyond this task. None is a new principle; each is a known one confirmed and bounded in a setting where the truth is in physical units. First, fit a combiner in the space where its errors live. Regression combination and stacking (Granger & Ramanathan, 1984; Breiman, 1996) fitted in log space lower error by 46% over the best source here and by 28% over the log-space constant they contain, while the same stack in cubic centimetres does not beat the best source. The constant's own 24% is a warning of another kind: where sources fail grossly, a combiner's gain must be read against a constant, not only against the best member. For the metric-scale cues, the reading goes the other way: what a learned combination gains is a fitted bias on one cue. Second, anchor on a strong prior rather than pool with it. The within-class regression read as a fusion rule keeps the prior's calibration and uses the source's ranking, and on ECUSTFD every combiner given more freedom does worse. The held-out test shows what that constraint is worth and when: on 74 targets it beats the combiners that nest it, and on $1{,}738$ it only ties the best of them, which can afford its free parameters. The gain is small, and when a class holds two or three targets, the slope is not worth estimating, as the condition $r^2_{\mathrm{wc}} > 1/n_{\mathrm{eff}}$ predicts. Third, the constant-source control is cheap and decisive: on ECUSTFD it shows the VLM's contribution to the chain to be shrinkage. None of the three needs a volume, a depth map or a camera, and we expect them to read the same way in any decision-level problem that pairs a strong prior with a weak measurement of the same quantity: a catalogue weight beside a load cell, a population rate beside a wearable, a prior map beside a range sensor. We have not run such a problem, so this is an argument from the estimator's form, not a demonstration. It recommends fitting the combiner in the space the errors live in, anchoring on the prior, fusing only the part of the source that varies within the prior's own class, and checking what a constant partner would have achieved before crediting a source with the gain. The released scripts compute all three from per-item predictions and labels alone. Per-sample selection remains open. The oracle removes 37–76% of the error on the all-channel pools, and a selector given the sources' own outputs recovers only part of it.

J.4 Limitations in full

The benchmarks limit what we can conclude. Only one volume dataset, MetaFood3D, passes every applicable criterion, and it does so on 34 items with a SKILL verdict that depends on the metric, so the strongest statements here are about evaluation rather than about which pipeline is best. No volume benchmark has both enough targets for an interval and real variation within its classes, so the threshold between benchmarks the category explains and those it does not is bounded across datasets rather than measured within one. MetaFood3D provides no physical scale cue, so part of what is measured there is each backbone's native scale, and the component check in App. D.7 covers only the four datasets that ship depth and masks. The support channel is a fusion axis only on NOCS and YCB: on food, a single segmenter was run, although segmentation varies widely on food imagery, and more so under viewpoint change (Alahmari et al., 2024; AlMughrabi et al., 2026). The instance identity behind every fold and interval proxies for the distinct ground-truth value, so two objects that share a volume are grouped together, which errs toward wider intervals.

The analyses carry their own limits. The fusion study reuses cached per-item predictions, so it aligns pools after the fact rather than producing them in one joint run. The primary family was fixed before the paired analysis but not pre-registered, and several analyses were motivated by the results they report: the class-anchored rule, its tuning (a clip at three robust standard deviations, the class median as the prior and the global fall-back, all set after the degenerate ECUSTFD item was seen), the multi-source extension and the scale-cue analysis. The rule itself was afterwards confirmed on held-out samples under a registered plan (Sec. 5.6); its multi-source form was registered too and did not replicate on its primary pool (Sec. 5.6); the tuning choices and the scale-cue analysis carry permutation, partition and constant controls, but no confirmatory weight. The gain itself is small: about 6% of the log RMSE on the paper's ECUSTFD split. In mean log error, the metric the abstract and the confirmation use, it is 6–11% there, 13% on the held-out objects and 3% on the held-out dishes. The four backbones share a segmenter, a scale rule, and an operator, so their agreement is not four independent tests.